

## **General Disclaimer**

### **One or more of the Following Statements may affect this Document**

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Draft

Interim Report - "A Study  
of Software Management and  
Guidelines for Flight Projects:  
NASA Contract # NASW-3350

(NASA-CB-104525) A STUDY OF SOFTWARE  
MANAGEMENT AND GUIDELINES FOR FLIGHT  
PROJECTS Interim Report (Draper (Charles  
Stark) Lab., Inc.) 182 P HC A09/ME A01

N81-26744

Unclass  
15023

*TM: MC*  
*Code: DP-4*



**The Charles Stark Draper Laboratory, Inc.**  
Cambridge, Massachusetts 02139



May 9, 1980

Enclosure #2

Draft

Interim Report - "A Study  
of Software Management and  
Guidelines for Flight Projects"  
NASA Contract # NASW-3350

May 9, 1980

<u>Table of Contents</u>	<u>Page</u>
1.) Introduction	1
2.) Purpose	1
3.) Scope	1
4.) Approach	2
5.) Study-Subject Characterizations	5
5.1 JSC/Shuttle	6
5.2 MSFC/Space Telescope	8
5.3 GSFC/Landsat D	11
5.4 JPL/Galileo	15
5.5 TRW	20
5.6 SDC	24
5.7 Air Force - ESD	27
6.) Summary	30
 <u>Appendix</u>	
A-1.) Expanded Study-Subject Characterizations	32
A-1.1 JSC/Shuttle	33
A-1.2 MSFC/Space Telescope	45
A-1.3 GSFC/Landsat D	55
A-1.4 JPL/Galileo	66
A-1.5 TRW	85
A-1.6 SDC	109
A-1.7 Air Force - ESD	129
A-2 Listings of Interviewees and Prospective Workshop Participants	168
A-3 Interview Format	170
A-4 Checklist	177

## 1.) Introduction

The following constitutes an interim report on NASA Contract NASW-3350, "A Study of Software Management and Guidelines for Flight Projects." The contract calls for a survey of present software development policies and practices - Task #1; an analysis of these policies and practices - Task #2; and a final report recommending possible improvements - Task #3. This report summarizes the Task #1 and Task #2 efforts to-date, essentially providing the background information necessary to assess the adequacy of present NASA flight software development approaches. Further analysis and refinement of this information, partially as a result of a Workshop involving key NASA software practitioners, is expected to form the basis upon which preliminary improvement recommendations can be developed for NASA review.

## 2.) Purpose

The motivating NASA goal for this study was to enhance the timely and cost-effective production of reliable flight software. The immediate objectives of this study, in support of that goal, are to refine NASA-wide understanding of flight software development successes and problems and, as a result, develop management policy improvement recommendations that can be broadly agreed to within the NASA, particularly at the Center Directors' and practitioners' levels.

## 3.) Scope

The software upon which this study is focused is described below (a slightly modified version of the present NMI 2410.6 Scope).

"These requirements shall apply to all of the software for a selected flight program--whether it is onboard software, support software, ground software necessary to perform the mission operations or software used to conduct the major system level testing--that could, in any way, cause mission failure; seriously degrade the mission objectives; or adversely affect personnel safety.

Ground software that is institutional and multiuser, or part of a generic capability shall be identified as to its use in the Program, where it is documented, and how it is managed, but it is not controlled by this NMI. However, all mission peculiar or mission unique changes which must be made to institutional software to support a flight mission shall fall within the scope of these instructions."

The recommendations that eventually will result from the survey and analysis tasks reported on herein will emphasize NASA Headquarters-level, policy-oriented suggestions (eg., those that might improve the NMI 2410.6). However, additional recommendations will address the broader class of technical, educational, and managerial issues, including identification of potential follow-on efforts.

To further characterize the scope and focus of this study, it may be useful to contrast those subjects that typically will be addressed with those that, at least within this initial effort, will not be addressed.

o Subjects that will not be addressed:

- Requirements or design specification methodologies or tools
- Coding or testing techniques, tools, or facilities
- Modeling or simulation technology or facilities
- Programming languages, operating system technology, or support software

o Subjects that will be addressed:

- Policy for management of software development....what to do, not how to do it.
- Impact of the abstract and complex nature of software.
- Alleged inadequacy of management policies, procedures, planning, standards, control, focus, visibility, etc.
- Evolutionary nature of requirements, iterative nature of process, and contradictory nature of demands.

#### 4.) Approach

As implied by the previously stated objectives, the present study is intended to be a fact-finding effort. While certain of the resulting recommendations may be of immediate benefit, it is anticipated that identification of potentially fruitful avenues for future efforts will be equally important.

Several study assumptions seemed inescapable from the outset. For instance, a significant degree of operational autonomy for each of the NASA Centers was assumed to be desirable, if not unavoidable. Therefore, participation of major NASA software implementation Centers in this study was considered mandatory. Dialogue amongst these key NASA elements was considered to be essential if the management policy recommendations that

result are to be effective....either in the view of managers who must enforce the policies or developers who must abide by them.

The role of the C.S. Draper Laboratory is seen primarily, therefore, as that of a non-aligned organizing agent, intermediary, catalyst, and reporter.

At the direction of NASA Headquarters, a group of NASA flight software Centers/projects were selected as study-subjects (JSC/Shuttle, MSFC/Space Telescope, JPL/Galileo, GSFC/Landsat D). In addition, another Government agency (Air Force - ESD) and two Industry representatives (TRW and SDC) were included to capture outside experience. Many additional sources could be identified but this would, it was felt, unacceptably limit the degree of analysis each received. Section A-2 of the Appendix lists the interviewees and prospective Workshop participants.

The order of events for this study were as follows:

- o Preliminary review of each study-subject's policy documentation ....to initiate understanding of present policies and to prepare for the survey interviews to follow. A checklist of topical areas of potential importance to the survey was developed to serve as a common baseline. (See appendix, A-4).
- o Study-subject interviews....to solicit personal experiences and suggestions, and to obtain answers to generic and study-subject-specific questions. An interview format was generated to help focus these sessions. (See appendix, A-3).
- o Interim Report preparation....refining and documenting the information obtained from the above reviews and interviews. In addition, a draft listing of generic lessons-learned has been extracted including successful practices, potential problem areas/deficiencies, useful observations, potential recommendations, etc.

We are here now!

- o Workshop/Seminar....with the multiple objectives of acquiring concise identification, by experts, of key software development deficiencies and potential corrective measures, and providing them the opportunity for informal dialogue. (Scheduled for May 29th and 30th at MIT's Endicott House).
- o Final Report preparation....converting the draft observations and Workshop experience into specific recommendations for review by a NASA peer group prior to publication.

Participation by C.S. Draper Laboratory individuals on the Space Telescope, Galileo, and Landsat D software audit teams last year provided insight into the valuable "evaluation" aspects of the present NMI 2410.6 audit process. This study avoids duplication of those program-specific audit activities in order to encourage candor during the survey of present issues. The organization of the Interim and Final reports avoids correlation of identified problem areas with specific study-subjects. Rather, a separate Final Report listing of generic problem areas will complement the non-judgemental, study-subject specific characterizations of this report.

As a final note, concern has been raised on several occasions that policy recommendations generally draw too heavily on lessons learned from the past, overlooking the demands anticipated in the future and leading, in our case, to policies for the 70's rather than the 80's. We believe our approach addresses this issue constructively and it is a key item in the draft list of preliminary observations. In addition, we plan to discuss the issue at the Workshop.



## 5.) Study-Subject Characterizations

The Survey and Analysis tasks of this Study included a review of selected study-subject policy documentation and follow-up interviews with study-subject software managers and practitioners.

The following sections characterize each study-subjects' software development process by briefly describing their present policies, procedures, responsibility and authority assignments, review and control mechanisms, documentation, standards, etc. The appendix (section A-1) contains an informal collection of more detailed background information on each study-subject in the varying styles of each reviewer.

It should be noted that these descriptions are limited in both context and completeness. It would be misleading to draw comparative conclusions based on them due to the varying emphasis of each reviewer. We solicit expansion, correction, and clarification, particularly by those study-subject personnel who have not yet had the opportunity to review the material.

However, our primary thrust, especially during the interviews, has been on determining which policies and practices the study-subjects felt were or would be useful, NASA-wide, rather than on detailed explanations of their present operations. Therefore, the non-judgemental, study-subject specific content of this Interim Report serves primarily as background material for the generic, (i.e. non study-subject specific) judgemental Final Report recommendations.

The Study-Subject characterizations that follow are listed below:

- 5.1 JSC/Shuttle
- 5.2 MSFC/Space Telescope
- 5.3 GSFC/Landsat D
- 5.4 JPL/Galileo
- 5.5 TRW
- 5.6 SDC
- 5.7 Air Force - ESD

## 5.1 JSC/Space Shuttle

The software development process for the Space Shuttle Program is quite accurately represented by the policies documented in JSC 07700, Volume XVIII, Book 3, Software Management and Control, Revision 9, dated December 19, 1978. A generic model of the software development life-cycle is described within the framework of a system development process. The life-cycle is divided into phases: requirements definition, preliminary design, detailed design, coding, verification and integrated testing. Each phase (except coding) is followed by a project (e.g. Orbiter) level review. The required activities of each review are specified including which review-item(s) are to be placed under NASA configuration control. The final review results in end-item acceptance. The policy also calls for reviews at the software developers (e.g. JSC/SSD-IBM) level to review schedules, cost and performance for each important product and activity. These latter reviews are to be specified in the software development plan.

A three level configuration control process is described. At the program level, the Program Requirements Control Board (PRCB) controls the top-level computer system and software requirements (JSC 07700, Vol. XVIII, Books 1-3), the Master Verification Plan and the inter-project (e.g., Orbiter-SSME) systems-software interfaces. The Space Shuttle Program Manager is the Chairman of the PRCB. At the project (Orbiter) level the Configuration Control Board (CCB) has the authority and responsibility to establish baselines for each software deliverable end-item; to approve implementation standards (e.g., programming languages) and test acceptance criteria; and to control changes, performance levels, costs, risks, and schedules. The chairman of the orbiter CCB is the Director of the Orbiter Project office. The Orbiter CCB has delegated this responsibility to the Orbiter Avionics Software Control Board (OASCB) for the flight software. The chairman of the OASCB is the Director of the Orbiter Avionics Project office. The software developer (JSC/SSD-IBM) is responsible for supporting the project level configuration control process and for performing the detailed configuration management of the software designs, builds, tests, and related documentation.

Book 3 describes the configuration control process and a flow chart is included that illustrates the order of activities at all three levels (program, project, and software developer).

The generic types and minimum contents of the documentation required to support software development and verification is defined. There are twelve such documents identified and grouped into five categories:

- 1) Management: Management Plan, Development Plan, Quality Assurance Plan,
- 2) Performance and Design Requirements: Requirements Specification,
- 3) Design: Preliminary Design Specification, Detailed Design Specification, Programmer's Handbook,
- 4) Test Planning: Test Requirements, Test Plan, and
- 5) Testing: Development Test Specification, Software System Test Specification, Verification Test Report.

Books 1 and 2, while less pertinent to this study, are characterized below.

Book 1 of Volume XVIII, ALLOCATION OF COMPUTATIONAL FUNCTIONS, delineates the top-level computational responsibilities of each major computational element (Orbiter, Launch and Landing, Mission Operations, Main Engine) during the various operational phases (Installation and Checkout, Mate and Interface, Preflight, Flight, Post-Landing, and Flight Planning).

Book 2, ALLOCATION OF SIMULATION FUNCTIONS, defines the roles of major Shuttle simulators and allocates functions to them.

Finally, Book 3 has a section that generically describes the precursory system level information and documents that are required by the software development activity to design, develop and verify the software.

MSFC/Space Telescope

Two MSFC software management policy documents were the subjects of this study. They are: 1) MSFC SOFTWARE MANAGEMENT REQUIREMENTS FOR FLIGHT PROJECTS, MMI and 2) MSFC SOFTWARE MANAGEMENT AND DEVELOPMENT REQUIREMENTS, MA-001-006-2H.

The first is a brief (6 pages) top-level policy document that is in part identical to portions of NMI 2410.6.

The second is comprised of seven books that describe requirements in specific terms. The subject matter (titles) of the seven books are:

1. Overview and Guidelines
2. Software Configuration Control
3. Documentation, Deliverables, and Planning Data
4. Software Data Requirements
5. Software Standards, Procedures, and Quality Assurance
6. Parent Systems Development
7. Independent Verification and Validation

Neither of these documents is project specific. All policies and requirements are stated in generic terms and each project is free to adapt them to their specific environment provided they comply with the intent of the policies.

Like the NMI, the MMI requires (unless specifically waived) that all software used on complex, high-profile flight projects comply to these policies. The MMI also outlines the authorities of the offices and groups that have primary responsibility for management of software development. Of particular note: the procurement office is charged with insuring that policy provisions are included in all applicable contracts. The responsibility of a Peer Review Committee is also included.

The Seven-book MA describes the activities required in the software development from the conceptual phase through operations and maintenance. Book 1 briefly describes each phase: conceptual, requirements, design, code and debug, verification, validation, systems integration, and operations and maintenance. Guidelines for the degree of applicability of the policies are included in Book 1. The degree of applicability is based on the size, complexity and criticality of the project. Guidelines for the software reviews to be conducted at key development milestones are also included in Book 1. These reviews are: systems requirements review (SRR), systems design review (SDR), software requirements review (SRR), software preliminary design review (PDR), software critical design review (CDR), software test review (TR), functional configuration inspection (FCI), configuration inspection (CI), and acceptance (AR). These software review milestones are placed within the context of the parent system's development cycle.

Book 2 describes the software configuration change control process, including a flow diagram that depicts the required sequence of activities. The various forms used to document changes and discrepancies are also included and described. The responsibilities of three levels of control authority are outlined: the Program/Project Configuration Control Board (CCB), the Software Review Board (SRB), and the software developer's internal control.

Book 3 describes the Documentation, Deliverables and Planning Data for MSFC projects developing software. Twenty-four documents covering management, development, test, and operations are described and a documentation tree showing their interrelationship is included. Guidelines for the number and physical form of typical software deliverables are included. Six types of software planning data

are described: Software Function Description, Software Specification Document Tree, List of Computer Program Contract End Items (CPCEI's), Development Schedules, Support Facilities, and Management Summaries.

Book 4 contains a Data Requirements (DR) form for each of the twenty-four documents described in Book 3. The DR contains submittal requirements and a description of the document.

Book 5 describes the standards, procedures and quality assurance requirements to be used during software development. The standards and procedures cover topics such as: structured programming, internal software data and hardware tests, naming standards, input-output interfaces, Programmer Notebook, coding, and testing. The responsibility of the software developer's QA program is described. These responsibilities cover such areas as configuration management, testing, deficiency correction, and documentation. The QA function is required to be independent of lower level technical management.

Book 6 is a high level description of a typical parent system development process from conception through integration and acceptance. The relationship between the systems and software development life cycles is briefly described.

Book 7 describes a three phase Independent Verification and Validation (IV&V) process. Each phase consists of two parallel activities: Phase 1 covers requirements analysis and test planning, Phase 2 covers equation and design analysis and facility development, and Phase 3 covers coding analysis and testing. The IV&V documentation requirements and management review responsibilities are briefly described.

### 5.3 GSFC/Landsat-D

The Goddard Space Flight Center (GSFC) personnel interviewed were involved in three distinctly different flight software-related efforts. They were:

- 1) Flight software produced by an industry contractor as part of a larger spacecraft contract (Landsat-D);
- 2) Flight software produced internally by GSFC personnel;
- 3) Flight supportive ground software produced primarily by "captive" contractor personnel under GSFC technical supervision.

The management approach of each of these was quite different. Each of the categories will be discussed individually. All of the documentation reviewed as part of this study was used or generated in support of item "1)" above, the Landsat-D program.

### Discussion

1. The flight software for Landsat-D is being procured as part of a larger contract for the spacecraft and integration. The type of contract chosen was a function of system requirements and not a function of the requirements for software, hardware, integration, etc. individually.

Subsequent to contract award, negotiations took place to insure the adequacy of the management plans and their conformance to GSFC requirements. It has been stated that NMI 2410.6 considerations will need to be included in future RFPs to insure proposals are in compliance.

The General Electric SEAM document is the model upon which the Landsat-D software management is based. Figure 1 shows the SEAM flight segment software development process and is extracted from the SEAM document. The SEAM document defines this process in generic terms to enable application to a broad range of projects. The SEAM consists of four levels;

- 1) Policy - Top level statement of policy and goals;
- 2) General Instruction - Mandates software engineering and management policies using a top-level representation of the process model and a brief elaboration of items in the model;
- 3) Standard Outline - Provides skeleton outlines defining content of all documents supportive of mandated policies, and
- 4) Guidelines - Examples, annotated outlines and guides to support 2) and 3) above.

Interaction between GSFC and the contractor for both technical and management concerns occur at frequent intervals (on the order of every other week) and at periodic milestone reviews. The more frequent meetings are based upon the evolution of documents that support milestone reviews such as the "Requirements Baseline" and management concerns such as schedule and personnel availability. These meetings in general are involved with in-scope contract issues. The milestone meetings are a culmination of the previous meetings and involves management in the approval of technical documents and resolution of contract issues.

The contractor responsible for development is also responsible for operations and maintenance over the first 90 days of the spacecraft mission. A separate contract will be let for subsequent operations by open competition.

A review of the Landsat-D documentation indicates most of the concepts in NMI 2410.6 are addressed.



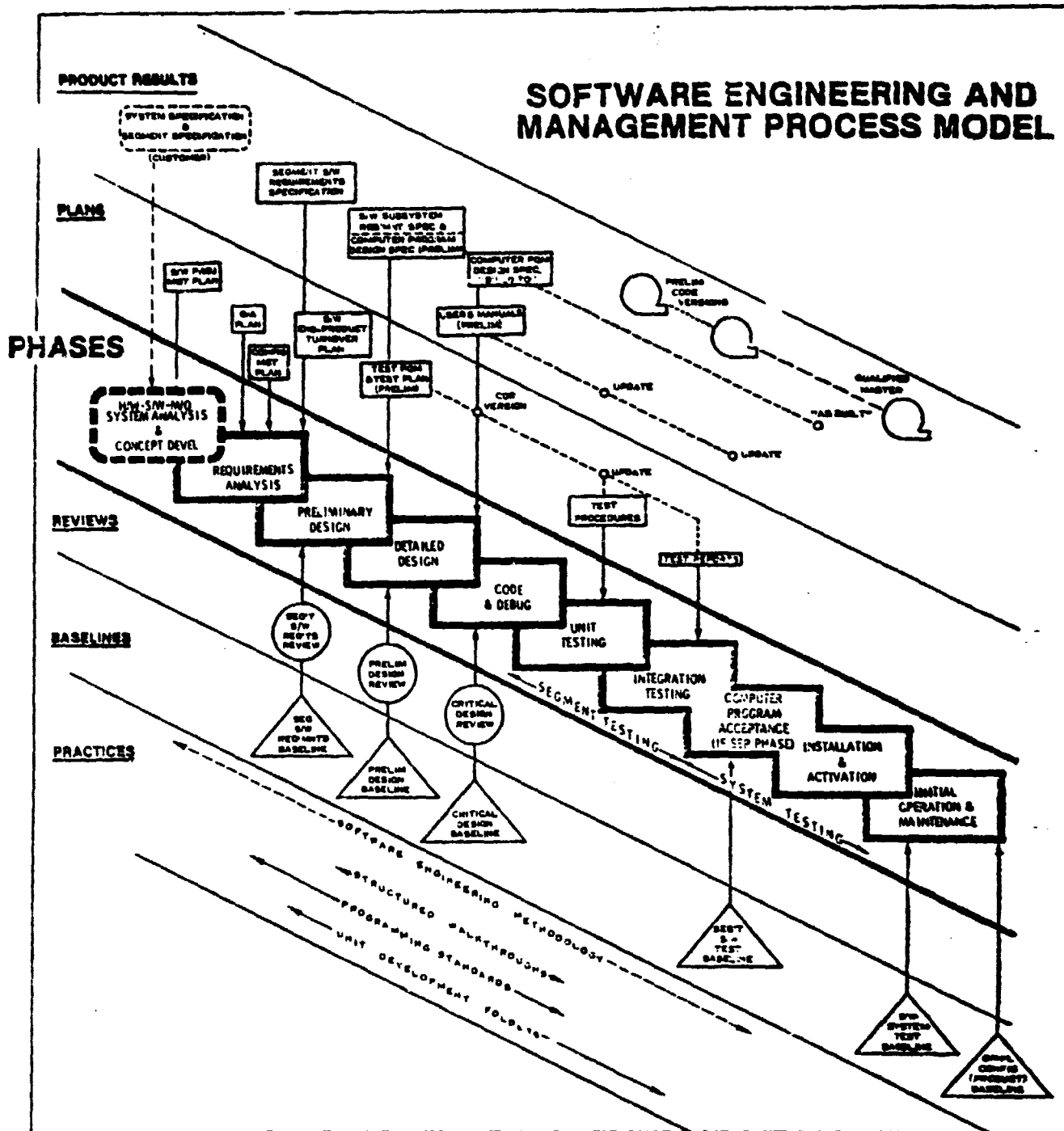


Figure 1. Top-Level Representation of the SEAM Process Model for Software Development Projects

2. GSFC in-house software development has been accomplished for programs such as the "Orbiting Astronomical Observatory (OAO)" and the "International Ultraviolet Explorer (IUE)". The size of the software task has grown with each successive program and while management controls and constraints on software development have been for the most part informal, an effort has been made to tailor the management policies to fit the task at hand. The OAO program was small enough to allow a close informal working relationship between programmers, providing free flow of technical information, but little management visibility. The software requirements/specification document grew as the software was produced and had no formal controls. The IUE program which followed was somewhat larger and more complex and justified a more formal management atmosphere. A formal requirements document was generated prior to the start of coding and was the responsibility of the systems engineer. The document was controlled by a "mini" configuration control board. This approach provided visibility and opportunity for management control.

It is understood that a means of implementing NMI 2410.6 in the GSFC internal environment is in process.

3. The flight supportive ground software discussed has been used for non-real-time applications (several minutes delay) such as spacecraft attitude determination and determination of maneuver and flight control information. The software was produced by contractor personnel, procured under a level-of-effort contract, under the technical supervision of GSFC personnel.

The contractor personnel had imposed upon them a management policy derived by the segment of GSFC responsible for the software being produced. The policies are documented; however, the imposition of the policies is by mutual agreement and not by contract. In this case the policies have evolved during the period over which the same contractor has been used.

The present NMI 2410.6 does not apply to the software development; however, the responsible GSFC personnel feel that most of the concepts in NMI 2410.6 are addressed in their policies.

#### 5.4 Jet Propulsion Laboratory/Galileo

The Project Galileo Software Management Plan, Volume I, Technical Development and Management Approach (1979), is up to date and matches the actual software development process as described during our interview at JPL. Software development is made difficult because it originates from 4 sources: 1) new software developed for Galileo, 2) software inherited from Voyager, 3) Mission Control and Computing Center (MCCC) Software with its own policies, and pre-existing software, and 4) Deep Space Network (DSN) Software with its own policies and pre-existing software. Chapter 5 of the Plan specifically addresses the exceptions to the Plan necessary for realistic development of a project in the JPL environment with this pre-existing software. Figure 1-1 shows the Plan Organization Overview with its seven life-cycle phases: Requirements Development, Preliminary Design, Detailed Design, Coding and Debugging, Testing, Integration and Delivery, and Maintenance.

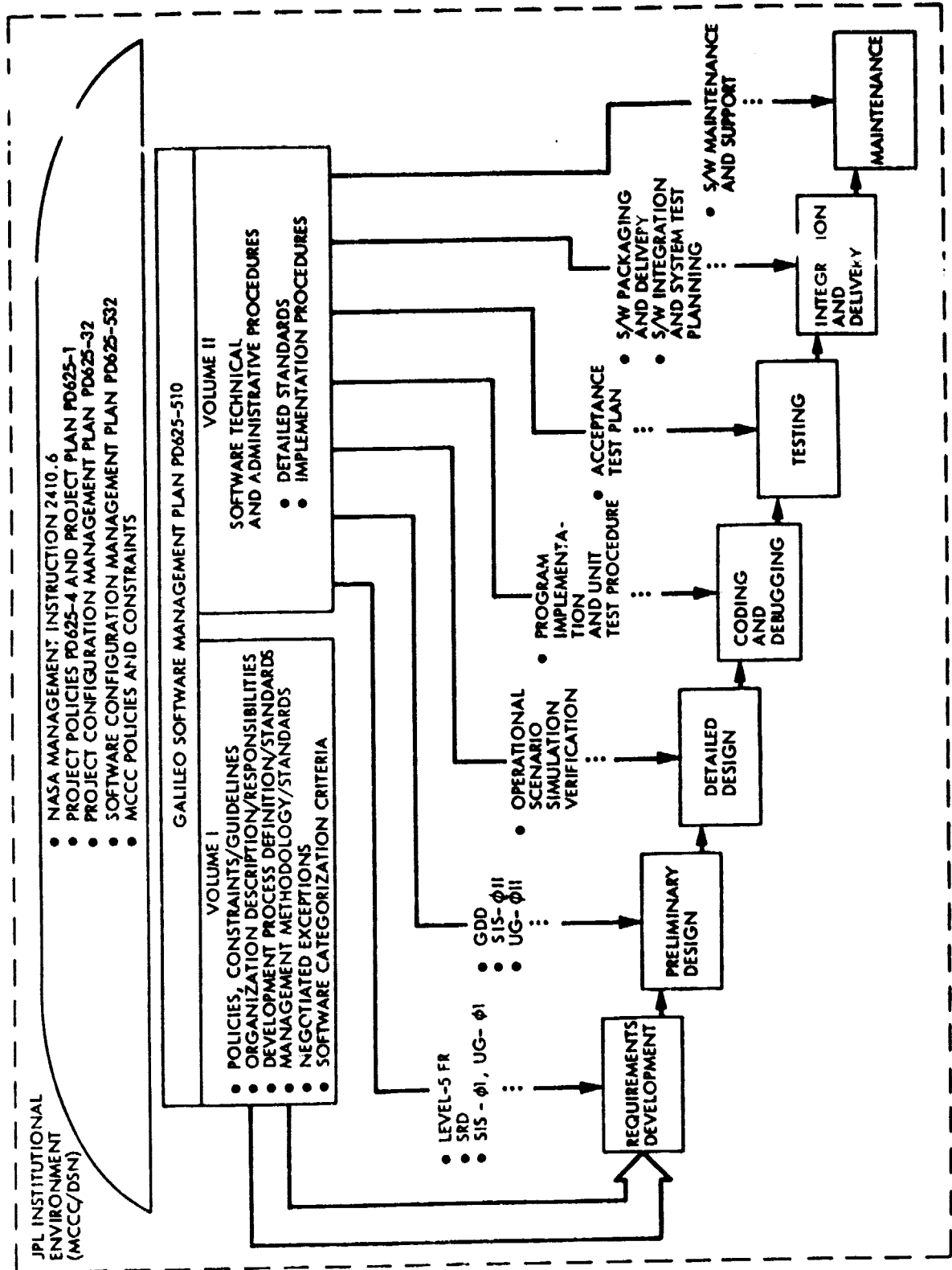


Figure 1-1. Plan Organization Overview

Thirteen Milestones/Technical Reviews corresponding to these phases are detailed in that Plan, along with documents and responsible parties. Four of these Milestones specifically call for a schedule/cost analysis to update either the System for Resource Management of the Project Office or one of the Technical Division Software Implementation Plans.

During the Requirements Development phase the three Project System Functional Requirements Documents (Navigation, Orbiter, and Mission Operations) are divided by the System Engineers into Subsystems Functional Requirements with a corresponding Software Functional Description Document. The Integrated Software Functional Diagram is then generated. With that, the Cognizant Engineers generate Preliminary then Final Requirements Documents (SRD), User's Guide (UG), and Interface Specs (SIS). Next, while Cognizant Programmers generate the General Design Document in PDL for the Preliminary Design Review, in parallel the Cognizant Engineers generate the Acceptance Test Plan.

The General Design Document (GDD) is expanded for the Critical Design Review by the programmers, and a Phase II User's Guide and Interface Spec are completed. Coding proceeds by the Programmer in a top down manner and ends with unit testing. Three further levels of testing are specified. Acceptance testing by the Cognizant Engineer demonstrates the software to be a stand-alone product. Integrated Testing by the IV&V contractor demonstrates the software as an element of the subsystem and of the system. The last testing phase consists of end-to-end scenarios in a pseudo-realistic operational environment.

When the software is then transferred to Maintenance Phase, each component has an identified Sustaining Engineer and a Regression Test Baseline - one or two small test cases used in acceptance testing and available to check modified software or check proposed system level changes.

The Project Configuration Management Plan (PD625-32) defines four product baselines: Requirements (Built-To), Design (Code-To), Program (As-Built), and Regression Baseline (As-Built Compliance). A formal change control procedure is defined for software categorized as mission critical under direction of the Project Software System Manager, (or higher), via a Software Change Control Board. For other software, a local change control procedure is available to the Project Software System Engineer. A third change procedure is available to the Software System Engineer for emergencies in urgent situations.

The Plan additionally covers Flight Software Reprogrammability, Margin Management, Quality Assurance, and Portability.

Volume II of the Plan, Software Technical and Administrative Procedures (STAPs), is quite comprehensive. STAPs are accumulated in separate sections, usually just prior to each new step in the development process, under the responsibility of the Project Software System Engineer. STAPs cover detailed standards and implementation procedures and provide for a "common interpretation" of the plan.

The Deep Space Net (DSN) was also reviewed. Its historic practices are well described in Standard Practices for the Implementation of Computer Software (1978). The current revision, JPL Standard Practice, Implementation and Management of JPL Software (Draft-April 15, 1980, Preliminary), has the additional feature that management policies and detail required in the documentation products are different for each of four classes of software: 1) Critical to Mission success (or very expensive, or very large, or many users), 2) Significant to JPL task performance, 3) Important for organization effectiveness, and 4) Important for individual effectiveness (and inexpensive, and small, and used only by developer).

This section (5.5) is proprietary  
to TRW and requires their  
permission before distribution



## 5.5 TRW

The System Engineering and Integration Division (SEID) Software Development Policies (1977-1978) and the SEID Software Products Standards (1977-1978) are the policy and standards from which a project management plan is derived. Not unexpectedly, they closely match Air Force policies of the same time frame and, in fact, reference the MIL standard documents in which they are in compliance. A simplified overview of the Life Cycle and associated documents is attached as Figure 2-1. (The grand three-dimensional TRW lifecycle reproduces best in wall poster size).

Eight development phases are identified: Conceptual, Definition, Preliminary Design, Detailed Design, Development, Unit Test, Integration Test, and Acceptance Test.

The System Spec is generated during the Conceptual Phase. From that, the Definition Phase generates the Software Requirements Spec, the Software End Product Acceptance Plan, the Interface Spec, Data Requirements Plan, QA Plan, and Configuration Management Plan for Review at the Software Requirements Review (SRR). The Preliminary and Detailed Design Phases follow with the Preliminary and Critical Design Reviews (PDR and CDR) covering the Data Base Specification, the Design Spec, the Unit Test Plan, the Integration Test Plan, the Software Acceptance Test Plan, and the User's Manual. The Standards document contains examples of the required documentation and identifies which parts are mandatory.

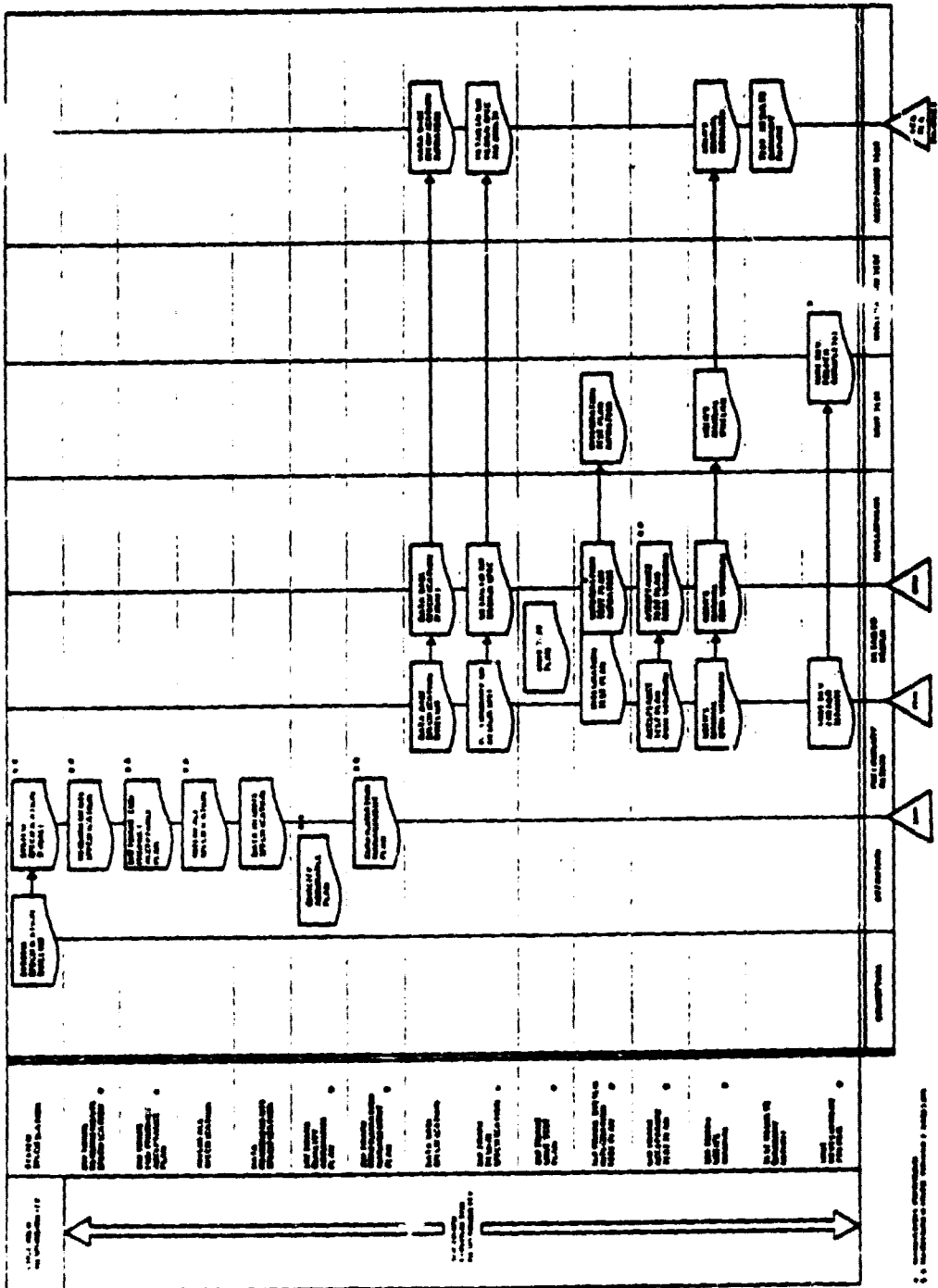


Figure 2-1. Time Phasing of SEID Software Documentation Set During Project Life Cycle

ORIGINAL PAGE IS  
OF POOR QUALITY

Unit Development Folders are the key management tool of the Development Phase. Procedures specify Top Down development, independent design check via a design walk-through, and guidelines for good programming. The three testing phases execute the three required test plans (unit, integration, and acceptance) and culminate with the Functional Configuration Audit, and the Physical Configuration Audit.

The functions of Configuration Management are specified in the Configuration Management Manual. Requirements for four basic functions are defined: 1) Identification, 2) Configuration Control Mechanisms, 3) Status Accounting, and 4) Configuration Verification. The degree of formality and control employed and manpower used is conditioned by the size and complexity of the project, the significance of the project, and the investment risks.

The Policy also specifies ongoing review and periodic quality audits that are to follow a Quality Assurance Plan. The Division Product Assurance Manager (PAM) is identified as responsible for the quality assurance functions and the configuration management functions. The QA Division is organized as a separate Division at TRW.

This section (5.6) is proprietary  
to the System Development Corp.  
and requires their permission  
before distribution.

## 5.6 System Development Corporation

In January of 1966, the System Development Corporation (SDC) issued a Software Development Manual (SDM), one of three components of their Software Factory approach to software development. The other components are a set of computer-based tools, and an organizational philosophy and discipline. The SDM contains a statement of corporate-approved policy for software development, and a phase-by-phase description of software products and activities.

The six SDC software development phases are:

- Planning
- Requirements/Performance
- Design
- Development
- Test and Acceptance
- Operations and Maintenance

In addition, the SDM describes phase independent activities: project and contract management, configuration management, and quality assurance. The basic documents produced during software development include a project plan, a computer program (CP) performance specification (requirements), a CP design specification (preliminary design), a CP detailed development specification (detailed design), a programmer's reference manual, a version description document, and test and user documentation. Besides the usual design reviews, the SDM calls for separate explicit reviews of the project plan, and the CP performance specification.

The SDM stresses the use of top-down design and development, implementation by successive builds, and reliance on a program production library for configuration management.

Our discussions at SDC confirmed that the SDM is a reasonably up-to-date description of actual practices. The manual is used more for reference than for training. It is also useful in describing their system to management within SDC, and to potential customers.

## 5.7 Air Force, Electronic Systems Division

Between 1975 and 1978, the Air Force Electronic Systems Division (Air Force Systems Command) contracted for the development of a Software Acquisition Management Guidebook (SAMG) series for command, control and communications (C<sup>3</sup>) systems. Additional guidebook series are being developed for flight simulator and avionics software. The C<sup>3</sup> series consists of 16 volumes covering such topics as documentation, verification, validation, life cycle events, contracting, maintenance, and software cost estimation (See Appendix 1.7 for a complete list). They were written by the MITRE Corporation and the System Development Corporation.

The intended audience for the series is Air Force program office management personnel who are responsible for managing software acquisition. The information presented includes material to supplement the official standards documents (lessons learned, common mistakes, etc.), checklists and descriptions of proven techniques, and references to the official directives and regulations pertaining to the topic under discussion.

The practices described are rooted in the DOD system acquisition life cycle which has a well-defined set of phases and milestones (reviews and audits). The basic phases are (see Figure 1):

- conceptual
- validation
- full-scale development
- production and deployment

Essential to the process are a set of specification documents, first at the system level, then at the configuration item or component level.

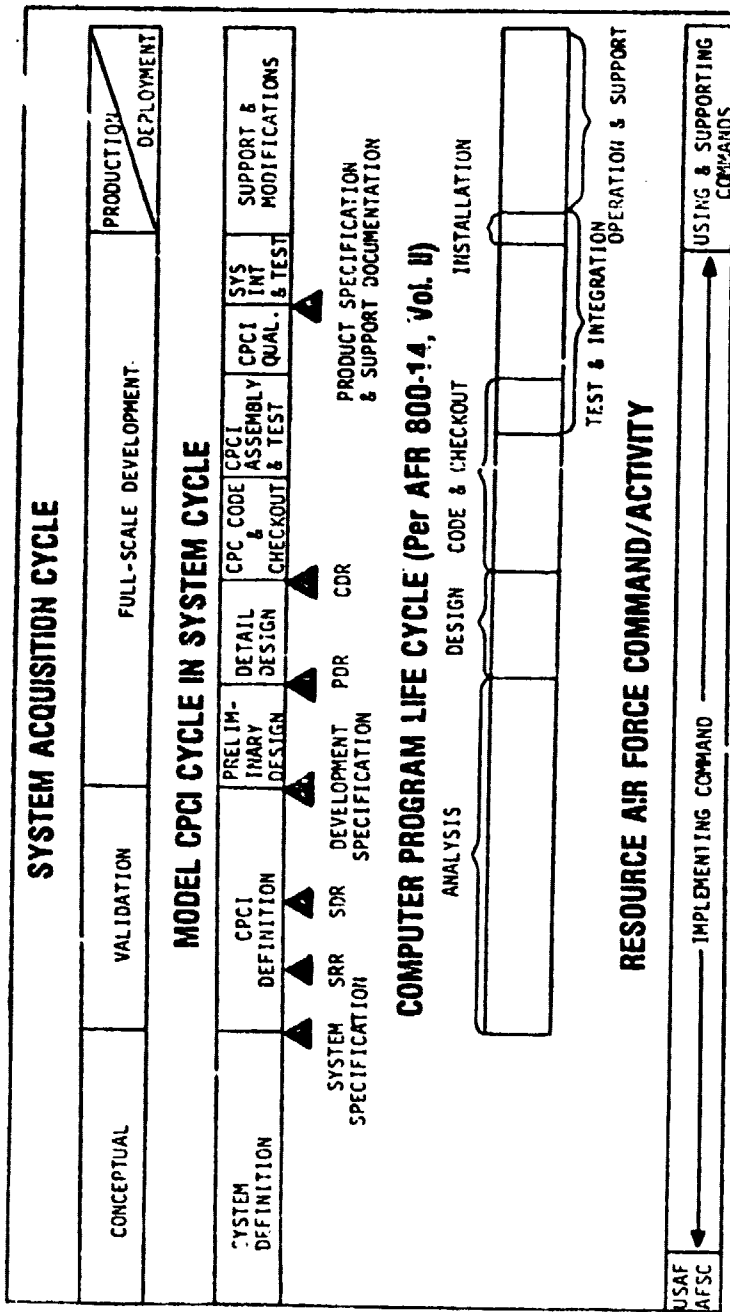


Figure 1. Summary of System Acquisition, Model CPCI, Computer Program Life Cycle, and Command Responsibility



Computer programs have been designated configuration items within the framework, and component specification documents (requirements and design) have been defined to govern their acquisition and configuration management.

These and other documents are reviewed and authenticated at various formal tones, such as the preliminary and critical design reviews (PDR and CNR).

The guidebook series presents a coherent picture of this evolving body of standards and practice.

6.) Summary

Each of the study-subjects has what appears to be effective software management and development practices in place. Though they differ slightly in style, form, and level of specificity, they all address roughly the same area of concern and follow generally the same life-cycle patterns.

Some of these differences, particularly between NASA Centers that often must work together on software projects, could perhaps, constructively be eliminated or clarified. However, these variations of terminology, style, form, etc, reflect the differing organizations and "ways of doing business" that goes along with the high degree of (and evidently desirable) Center autonomy. This issue, ie, desirable degrees of Software Management Practices commonality within NASA, will be discussed at the Workshop and in the Final Report.

## APPENDIX

## A-1 Expanded Study-Subject Characterizations

This background information has been gathered from documentation reviews and study-subject interviews. It is written in the individual styles of the various study team reviewers, though the enclosed Interview Format 3) and Checklist (A-4) formed a common baseline of sorts.

The Study-Subject Expanded Characterizations that follow are listed below:

- A-1.1 JSC/Shuttle
- A-1.2 MSFC/Space Telescope
- A-1.3 GSFC/Landsat D
- A-1.4 JPL/Galileo
- A-1.5 TRW
- A-1.6 SDC
- A-1.7 Air Force - ESD

## A-1.1 Characterization of JSC Shuttle Software Management and Control Policy.

The Space Shuttle Program Level Requirements are documented in JSC 07700. Volume XVIII of this nineteen volume set is the Computer System and Software Requirements. The writing of Volume XVIII was undertaken in late 1972 by the Data Systems Analysis Directorate of JSC in response to a request by the Program Director. Book 3 of Volume XVIII, SOFTWARE MANAGEMENT AND CONTROL, is the study subject of this effort, as it defines the program level policy for Shuttle Software Development. Book 3 has been revised ten times since official publication in 1974. Revision 9, dated December 19, 1978, was used in this study (revision 10 is a small editorial correction).

As previously noted, JSC 07700 is a program level document and is intended to apply to all Shuttle software elements, however this study was restricted to the policy as it applied to the Orbiter prime system flight software. The requirement to adhere to this policy has been included in the software development contract with IBM since the inception of the contract. The program has an active policy to maintain JSC 07700 as evidenced by the fact that ten page changes to Book 3 have been issued since its initial publication.

An examination of Book 3 vis-a-vis the checklist of this study revealed substantial agreement. The policy adequately covers most of the categories of the checklist. The same is true of the policy vis-a-vis the contents of NMI 2410.6. The characterization of this document follows the categorical elements of the checklist.

### 1. Top Level Policy

JSC 07700, Volume XVIII, Book 3 is the Software Management and Control policy document at the Space Shuttle Program Manager level. All elements of the Space Shuttle Program

(NASA and contractor organizations) must adhere to the policy. Book 3 describes the generic software development process, the configuration control process, and specifies the control responsibilities, documentation, and review milestones.

The following software reviews are required.

- 1) The Software Design Requirements Review (D&R) is held prior to initiating formal preliminary software design.
- 2) The Software Preliminary Design Review (PDR) is held to review the preliminary (functional) design before detailed design proceeds.
- 3) The Critical Design Review (CDR) is held to review the CODE-TO design, test acceptance criteria, and to place the detailed design under configuration control.
- 4) The Configuration Inspection (CI) is held to perform the following:
  - . Review changes from CODE-TO to AS-BUILT software design
  - . Review compliance of AS-BUILT software to requirements
  - . Review verification test results
  - . Place AS-BUILT software under configuration control
  - . Release software for systems and integrated tests (note: pre-CI test releases are permitted by waiver).

- 5) The Acceptance Review (AR) is held at the systems level to perform the following:
  - . Review the software changes since CI and related test results
  - . Review the system and integrated acceptance tests
  - . Establish acceptability of the software.
- 6) Incremental Reviews can be held when deemed appropriate in order to place portions of the software and documentation under configuration control.

The policy specifies the end-purpose of each review and delineates the documents and products to be reviewed. It also specifies the minimal contents of a Review Summary Report that must be published immediately after each review. A figure that illustrates the major phases, elements and milestones of the software development process is included.

## 2. Software Management Plan

JSC 07700 establishes the requirement for a Software Management Plan to be produced early in the software development cycle.

Plan objective: establish specific management policies and define the means by which these policies will be implemented.

Plan shall:

- . Define specific configuration control plan and the process(es) to be followed for defining the procedures for baseline definition, design review, review item disposition (RID), change control and change evaluation and reporting
- . Define approval authority for software milestone reviews (DRR, PDR, CDR, CI)
- . Establish the charter and responsibility of all organizations, boards, panels and working groups, including functions and interfaces of each plus functions associated with management of each. A work breakdown structure (WBS) will be included
- . Provide an overview of facilities required and define responsibilities for facility planning

- . Define specific documents required for software development and responsibilities for preparation of same
- . Establish management policies and responsibilities for ICD development, baselining, technical coordination, and review
- . Define roles and responsibilities for establishment and control of all Q.A. requirements
- . Define status reporting procedures to provide visibility of progress, including problem identification and disposition.

### 3. Configuration Management

JSC 07700 contains an excellent, extensive section (16 pages) on Configuration Management. All terms used are generic and well defined. A generic model (flow diagram) of the configuration control process is included. The organizations responsible for the various levels and stages of configuration control are identified. Three levels are identified: Program, Project and Software Developer's Internal Control. The stages identified are the DRR, PDR, CDR, CI and AR. The documents that are subject to configuration control are identified.

This section contains a requirement (section 3.6) that the software developer's organization shall have a Quality Assurance function that is not subject to direction from lower level technical management (which it shall audit).

### 4. Quality Assurance

JSC 07700 cites the requirement for a Software Quality Assurance Plan (section 4.2.3) The purpose of the plan is to ensure that uniform QA requirements are imposed during software development and test. The plan shall: Establish procedures for fulfilling all software QA requirements, define the QA review and reporting procedures, and establish auditing procedures for 1) design, coding and verification standards, 2) test procedure, and 3) tape/card handling.



5. End-Product Acceptance

JSC 07700 describes a two-stage acceptance of the software. The first stage is approval of the software for release for systems and integration testing. This is performed at the Configuration Inspection (CI) based on the review of software verification test results and the AS-BUILT Detailed Software Design Specification (see section 3.2.4). The first stage can be augmented with incremental reviews (see section 3.2.6). The second stage is the Acceptance Review (AR). This encompasses a review of the software performance during integrated systems testing plus a review of all software changes and corresponding verification test results since CI (see section 3.2.5).

JSC 07700, Vol, XVIII, Book 3, Software Management and Control (the book under review) states that the Software Management Plan shall define the approval authority and review procedures for the CI.

The AR is a system level review and is performed in accordance with JSC 07700, Vol. IV.

6. Operation & Maintenance Plan

JSC 07700 does not refer to such a plan. Operations is covered by: 1) a User's Guide, sect. 4.4.2.2 (part of the DSDS), 2) the System Operating Manual; sect 5.7.1 and 3) FOD generates a Mission Operations Plan for each mission/mission-type.

Maintenance is perhaps best covered by the configuration control requirements outlined in the Software Management Plan (sect. 4.2.1) and in the Software Developer's Internal (Configuration) Control (section 3.3.3).

7. Resource & Schedule Estimation, Monitoring and Control.

This topic is addressed in the Software Management and Development Plan Sections.

The Management Plan must define the status reporting procedures necessary to provide visibility of progress, including problem identification and disposition. It must also

define control and approval responsibilities.

The Development plan must contain the schedules, activities and resources required for software development.

8. Risk Assessment

The project level CCB has the responsibility to minimize development risks, but there is no specific focus on the ways or means to assess risks.

9. Documentation

JSC 07700 describes twelve development documents. These documents are typed as to the level of NASA configuration control. Type 1 documents must be approved by the project level CCB prior to implementation. Type 2 documents do not require formal NASA review and approval, but must be reviewed and approved by the organization responsible for generating the document. The twelve documents (more fully described elsewhere in this report) are:

- . Software Management Plan
- . Software Development Plan
- . Software Quality Assurance Plan
- . Software Requirements Specification
- . Preliminary Software Design Specification
- . Detailed Software Design Specification
- . Programmer's Handbook
- . Software Test Requirements
- . Software Test Plan
- . Software Development Test Specification
- . Software System Test Specification
- . Software Verification Test Report

All are type 1 except the Programmer's Handbook, which is type 2.

10. Development Plan

JSC 07700 calls for a Software Development Plan the objective of which is to present the schedules, activities, resources, and milestones required for the development of the software.

## 11. Requirements Specifications

JSC 07700 specifies the existence of three types of performance requirements specification documents.

### . System Requirements Specification

- . Purpose: Define functional, performance, and interface design requirements for each system.
- . Will include
  - . traceability of each requirement to source
  - . will address all system functional characteristics
  - . Preliminary computer characteristics
  - . System performance requirements in measurable/quantified terms with established acceptance criteria for each
  - . Identify operational requirements that impact hardware/software design
  - . Key assumptions and constraints in defining the external system interface requirements
  - . Systems quality assurance requirements.

### . Inter-Project Interface Control Documents (ICDs)

- . Purpose: Identify and control all interfaces that will affect the computer system and software design or operation across project lines.
- . Will include
  - . Definition of computer system interactions including:
    - . characteristics and conventions of interface signals and data
    - . functional descriptions of all commands that cross the interface: pre-command responses, timing requirements, and all information needed for implementation on both sides of the interface.
  - . Definition of interface signal validation, failure protection techniques, and error responses
  - . Operational sequence and constraints one (each) side imposes on the other
  - . Required support procedures (e.g., load, read-out) necessary for operation across the interface.

- . Software Requirements Specification
  - . Purpose: to establish software requirements for developer
  - . Shall contain:
    - . functional, interface and error recovery requirements in measurable/quantified terms. Must establish acceptance criteria for each requirement. Preliminary memory and timing requirements included.
    - . Key assumptions and constraints used in defining external software interface requirements
    - . Operational requirements that impact software design
    - . Software quality assurance requirements.

12. Design Specifications: Preliminary and Detailed

JSC 07700 cites the requirement for both a Preliminary and a Detailed Software Design Specification

. Preliminary SDS

The objective of the PSDS is to define the preliminary design approach in response to the Software Requirements Specification.

The PSDS shall: define design approach, provide traceability to the requirements specification, define design characteristics at software program level (e.g., sequencing, displays, errors, I/O, diagnostics, timing, memory, library use), define data base structure, and describe each function. Also it shall specify program structure, functions, interfaces, sizing and timing allocations, programming language and critical assumptions/formulations/constraints. In addition, it shall define equations, constants and include functional flow charts; and describe the hardware/software interfaces.

. Detailed SDS

The purpose of the DSDS is to describe software in sufficient detail to permit coding.

The DSDS starts as a CODE-TO spec that is reviewed and approved at the CDR (it is baselined and put under NASA CCB configuration control).

The DSDS becomes the AS-BUILT software description document after review and approval at the CI. The AS-BUILT DSDS is the CODE-TO DSDS with approved modifications.

The CODE-TO DSDS includes: requirements and design traceability, a program level description, a description of each module and a definition of module-to-module, software-to-hardware and data interfaces.

The AS-BUILT DSDS adds the following to the CODE-TO DSDS: A final program listing (source and object), a software user's guide, and a definition of operational limitations.

### 13. Program Development

JSC 07700 cites the requirement for a Programmer's Handbook (section 4.4.3). This document is to contain standards, guidelines, and system characteristics for software development, coding and verification. It includes, by reference, standards called out by JSC 08330, Computer System Hardware/Software Integration Handbook #1, Software Development Standards (to be used as guidelines only).

In addition, the handbook must define the procedures for implementing all standards, provide a brief description of available support tools, contain a list of reference documents required by the programmer and define additional data needed for design and coding. Finally, it must include testing standards identified in the Master Verification Plan, Volume IX.

### 14. Test, Evaluation, and Demonstration

JSC 07700 defines the requirement for five documents to directly support software level testing and four documents to support systems level testing.

The software level test documents are:

- 1) Software Test Requirements: objective is to establish test requirements to assure proper software function. Note: although this document is cited for NASA type 1 approval and control (Section 3.5.1, Table 3-1) and is depicted in the Software Development Process (Figure 3-1), it is not cited in the text as either an output of the requirement review (DRR) or as an input to the preliminary design review (PDR). Also, its review and baselining process is not described.
- 2) Software Test Plan: objective is to provide overview of test activities, schedules and resources necessary. Cited as necessary to support the CDR, but its review and baseline process is not described (see 1 above).
- 3) Software Development Test Specification: objective is to specifically define the test requirements for the software module and subprogram testing. Cited as needed to support the CDR, but no specification as to how it is reviewed and baselined (see 1).
- 4) Software Systems Test Specification: objective is to specifically define the requirements for software program and system testing. Cited as needed to support the CDR but no specification as to how it is reviewed and baselined (see 1).
- 5) Software Verification Test Report: this report shall summarize test results, identify unmet acceptance criteria, define test deviations and demonstrate compliance.  
Reviewed at CI.

The following system level documents are included in Sect 5 of JSC 07700, VOL XVIII, Book 3. They are stated as being required by the software development activity to design, develop and verify the software.

The four systems level test documents are:

- 1) Systems Test Requirements: purpose is to establish test requirements needed to exercise all hardware, software, and functions to the extent necessary to establish confidence in the complete system.
- 2) Systems Test Plans: purpose is to provide overview of test activities, schedules and resources necessary to perform testing through systems acceptance tests.
- 3) Systems Test Specifications: purpose is to define test requirements.
- 4) Systems Acceptance Test Report: summarizes acceptance test results, unmet acceptance criteria, test deviations and demonstrated compliance.

15. End-Product Use

JSC 07700 states the AS-BUILT Detailed Software Design Specifications shall include a software user's guide that: provides instructions concerning the use and options of the software, defines the diagnostic characteristics of the software along with maintenance procedures, and defines the program capabilities including function descriptions and logic diagrams.

JSC 07700 specifies that there shall be a Systems Operating Manual (Section 5.7). This is included in the section (5.0) for those documents that are needed for software development, but it is doubtful that this document can be generated before the software is fairly well developed.

16. Maintainance

Book 3 does not cover the software life cycle past acceptance, however, the required contents of the software user's guide includes maintenance procedures.

17. Productivity/Error Data Collection

Productivity data collection is not addressed. The only error data collection is provided by Review Item Disposition (RID) and Discrepancy Report (DR) forms.

18. Procurement

The general subject of procurements is not addressed in JSC 07700, Vol XVIII, Book 3. The software deliverables (media, documentation, data) are identified by reference to SN-5-0008, Space Shuttle Program Software Deliverable Data Package.

19. Certification

This word is not used. The QA function covers this area.



## A-1.2 Characterization of MSFC Software Management Documents

There are two MSFC Software Management documents.

- 1) MSFC SOFTWARE MANAGEMENT REQUIREMENTS FOR FLIGHT PROJECTS (MMI)
- 2) MSFC SOFTWARE MANAGEMENT AND DEVELOPMENT REQUIREMENTS,  
MA-001-006-2H

The first is a brief (6 pages) top-level policy document that is in part identical to portions of NMI 2410.6.

The second is comprised of seven books that detail specific requirements. It is a descendent of JSC 07700 (Volume 1, Book 3) and NHB 2410.1B (Appendix G, NASA SOFTWARE MANAGEMENT GUIDELINES).

The subject matter (titles) of the seven books are:

1. Overview and Guidelines
2. Software Configuration Control
3. Documentation, Deliverables, and Planning Data
4. Software Data Requirements
5. Software Standards, Procedures, and Quality Assurance
6. Parent Systems Development
7. Independent Verification and Validation

The MSFC policy documents are preliminary and a revision will be published in mid 1980. They are expected to be approved as official MSFC policy soon thereafter. The documents are being written by the Data Systems Laboratory and are based on NMI 2410.6, NHB 2410.1B, JSC 07700, and present MSFC software development practices. Two significant changes are planned for the mid 1980 revision: Book 6, Parent Systems Development will be deleted (a separate policy document covering this topic is planned), and the IV&V aspects will be rewritten to remove the impression that it is a separately contracted activity.

The MSFC documents describe a generic set of policies that apply to all projects, both contractual and in-house, and to all software, including flight, operational, ground checkout, support, and simulation. Each project is free to tailor the policy to its specific needs during the conceptual phase and the approved policy must be reflected in all applicable contracts.

This study's review of these documents revealed substantial agreement with the checklist. In addition, there is no substantive disagreement between the contents of NMI 2410.6 and the MSFC policy.

The characterization of these documents follows the categorical outline of the checklist.

A. Essential for Management

1. Top-Level Policy

The top-level policy is embodied in the MSFC Management Instruction (MMI) entitled "MSFC SOFTWARE MANAGEMENT REQUIREMENTS FOR FLIGHT PROJECTS." This MMI references NMI 2410.6, NHB 2410.1B and MA-001-006-2H. The SCOPE section of the MMI is a direct paraphrase of the SCOPE and APPLICABILITY sections of NMI 2410.6, i.e., the MMI applies to all software used on complex high-profile flight projects (save "generic" institutional and multiuser ground software).

The MMI outlines the responsibilities of the following offices and groups:

- . The MSFC Director has the responsibility of appointing a Peer Review Committee and of reviewing and approving the Software Management Plan. He also has the authority to waive the requirements for a Software Management Plan.
- . The Peer Review Committee has the responsibility of reviewing the Software Management Plan in accordance with the requirements of NMI 2410.6. The Committee continues its review until the Plan is satisfactory.
- . The Project Manager helps the Director determine if a Software Management Plan is required. If so, he is responsible for developing the Plan, utilizing MA-001-006-2H as a guide and for the implementation of the approved Plan.
- . The Office of the Associate Director for Engineering is responsible for coordinating the Software Management Plan activities within the Science and Engineering organization.
- . The Data System Laboratory is responsible for providing technical support in the implementation of the MMI; for review of and concurrence with the Software Management Plan, and for providing at least one member of the Peer Review Committee.
- . The Reliability and Quality Assurance Office is responsible for reviewing, critiquing and concurring in the Software Quality Assurance Plan.

- . The Procurement Office is responsible for insuring that provisions of the MMI are included in all applicable MSFC contracts.

The MMI includes two Matrix Flow-Charts that depict the organization responsibilities for each of the ordered steps involved in the development of an approved Software Management Plan. There is one chart for out-of-house (contracted) and one for in-house developed software.

The MMI also includes definitions of:

- . Complex High Profile Flight Program - any MSFC flight program that has a full-time Project Manager.
- . Peer Review Committee - an independent group of selected experts appointed by the MSFC Director, including up to three members nominated by the NASA Headquarters Office of Chief Engineer.
- . Software - a computer program resident in computer hardware.

#### Comments

The MMI is an excellent top-level policy. It is short and concise, and its application of generality vs. specificity is appropriate, i.e., the assignment of responsibilities to specific offices and groups without elaborating those responsibilities beyond the depth of the development of the Software Management Plan. The definition of a "complex high profile flight program" is laudable because of its pragmatism (any that is directed by a full-time Project Manager).

#### 2. Management Plan

Book 3, Documentation, Deliverables and Planning Data, outlines the required contents of the Software Management Plan. One possible improvement would be the requirement for the generation of a Software Facilities Development Plan.

The required contents are repeated, along with submission requirements, on a Data Requirement (DR) form in Book 4, Software Data Requirements. (Note: the abbreviation DR is used for both Data Requirement and Discrepancy Record).

### 3. Configuration Management

The entire contents of Book 2, Software Configuration Control, is devoted to this subject. The roles and responsibilities of three software control boards are described: the Program/Project Configuration Control Board (CCB), the Software Review Board (SRB) and the Software Developer's Internal Control.

The configuration change process is described and two flow charts that depict the change control process for new or expanded requirements and for changes to resolve discrepancies are included.

The nine different forms that are involved in the change control process are briefly described and a copy of each is included.

The configuration status accounting process is described. The status must be maintained in a data base (presumably computerized) and the coordination of all status information is charged to a configuration management librarian. A configuration chart which shows all end items and changes to those end items is required for each project.

Finally, Book 2 identifies three Document Type classifications, and defines the level of NASA approval and control for each type.

### 4. Quality Assurance

A Software Quality Assurance Plan is required and the MMI gives the Reliability and Quality Assurance Office review and concurrence responsibility. The purpose and required contents of the Plan are described in Book 3 and in Book 4.

The requirements and responsibilities of the Software QA Program are described in excellent detail in Book 7, Software Standards, Procedures, and Quality Assurance. Independence of the QA function is assured by requiring that it not be subject to direction by levels of technical management that it audits.

### 5. End-Product Acceptance

Book 1, Overview Guidelines, describes the test/review milestones that lead to end-product acceptance.

The Functional Configuration Inspection (FCI) occurs at the completion of verification and start of validation. It results in

placing the as-built software under configuration control.

The Configuration Inspection (CI) occurs at the completion of validation and prior to delivery of the software for systems integration.

The Acceptance Review (AR) results in acceptance of the integrated system, including the software. The Software/Systems Acceptance Test Specification defines requirements and procedures for testing.

#### Comments

This is a good generic model (and quite similar to that described in JSC 07700) that can be adapted to individual projects. The question of separate acceptance of the software (apart from the system) is not specifically addressed but it could be accomplished within the model if required, e.g., separate acceptance at CI.

#### 6. Operations & Maintenance Plan

Book 1 defines the Operations and Maintenance Phase (from AR to end of project) and states that the configuration control process must be maintained during it.

Books 3 and 4 also state the requirement for a Software Users' Manual. However, Books 3 and 4 differ in their description of the purpose and contents of the Manual. The Book 4 description is of an operator's guide or user manual whereas the Book 3 description is more fitting of a programmer's guide.

#### Comments

Assuming the Book 4 description of the Users' Manual is used, the Software Maintenance Plan and the Software Users' Manual fulfill the need for an O&M Plan.

#### 7. Resource & Schedule Estimation, Monitoring and Control

Both the Software Management Plan and the Development Plan require : call for resource and schedule monitoring and control procedures, guidelines and techniques to be implemented. In addition, a Software Schedules Document is required. This document is to contain the up-to-date, detailed schedules for all software and related documentation.

Book 3 also mentions certain software planning data. These are: Software Functional Description, Software Specification Document Tree, List of Software Computer Program Contract End Items, Development Schedules, Support Facilities, and Management Summaries.

The Management Summaries are to be compiled from the other planning data to show critical elements, schedule adherences, time and core monitoring, development costs, and management action items.

In addition, Books 3 and 4 call for a Software Status/Problem Report Plan the purpose of which is to provide guidelines and techniques for the implementation of a software status/problem report system. The status report provides the revision level, last compilation/assembly date, number of expended man-hours and a description of status and problems of each end-item.

8. Risk Assessment

This topic is not discussed.

9. Documentation

Books 3 and 4 describe the documents used in the software development. There are twenty-nine documents. Four are categorized as management documents, nine as development documents, nine as test documents, two as operations documents, and five as independent verification and validation documents. A documentation tree is included which shows how the twenty-four non-IV&V documents are related.

B. Essential for Development & Use

10. Development Plan

A Software Development Plan is required and its content is described in Books 3 and 4.

The Software Development Plan is initially submitted three months after the contract is awarded and updated as required.

11. Requirements Specifications

There are two requirement specifications; the Software Requirements and the Test Requirements.

The Software Requirements Specification is derived from (based upon) the Systems Requirements Specification. There is one document for each contract end-item. It is initially submitted one month prior to the Software Requirements Review (SRR). After the review it is baselined (put under configuration control). Updates occur as required.

The Software Test Requirements are published along with the Systems Test Requirements just prior to the SRR. There is one Software Test Requirement document per contract end-item. It is reviewed at the SRR and baselined (put under configuration control) and approval. Updates occur as required.

There are a number of System level requirements documents from which the Software Requirements are derived. They are:

- . Interface Control Documents (ICD)
- . Systems Requirements Specification
- . Systems Test Requirements

These documents are described in Book 6, Parent Systems Development.

## 12. Design Specifications

There are two Software Design Specifications; Preliminary and Detailed.

The Preliminary Software Design Specification defines the preliminary design approach in response to the Software Requirements Specification. It is reviewed at the PDR but not baselined. However, changes to the baseline do require approval of the Software Development Manager.

The Detailed Software Design Specification initially serves as a CODE-TO Spec for the programmers. It is reviewed and baselined at the CDR. The Detailed Design Spec evolves into an AS-BUILT spec. It is reviewed at the Test Review and at the Functional Configuration Inspection (at which the as-built software is placed under configuration control). It is subjected to a detailed audit at the CI. The AS-BUILT Spec consists of the CODE-TO Spec plus a final program listing and a software user's guide (Note: this is separate from the Software Users' Manual).

13. Program Development

This topic is addressed in three aspects. First, as a management document, a Software Standards and Procedures Document is required to be submitted three months after contract award. Some of the purposes of this are: (1) Define the standards and procedure requirements for design, coding, verification, and validation, (2) Define the detailed standards and procedures to be used, (3) Define the software products to which the standards apply, and (4) Identify any non-software standards and procedures that affect the software.

Second, a Programmer's Handbook is required to be submitted by the Systems CDR; one per computer/peripheral set. This document contains standards, guidelines and information concerning systems characteristics to be used by programmers for coding and testing.

Third, Book 5, Software Standards, Procedures, and Quality Assurance, describes the software standards and procedures for MSFC projects developing software. Many topics are included and some are quite detailed and specific. The topics include: Modular, Top-Down and Structured programming, Flow Charting, Error Checking, Coding, Debugging, Data Base Structure, Naming Conventions, Interfaces, Programmer Notebooks, Source Code Generation, and Testing.

14. Verification, Validation, Testing, Evaluation & Demonstration

This subject matter is addressed by a number of documents. The Software Test Plan provides an overview of the activities, schedules and resources necessary to perform software testing. The Test Plan is submitted three months after contract award and updated as required.

The Software Verification Test Specification describes each verification test case. Evaluation criteria must be included.

The Software Validation Test Specification provides guidelines and techniques to be followed in the validation of the software and defines the interrelationships that exist between documents that affect software validation.

A Verification Test Report and a Validation Test Report are required after each respective test phase for each end-item.



A Post-Flight Software Evaluation Report must be submitted by the contractor after each flight to aid NASA in evaluating the performance of the software during the mission.

Book 7 is devoted to the subject of Independent Verification and Validation (IV&V). IV&V is divided into three phases that span the software development life cycle. Phase 1 is requirements analysis and test planning. Phase 2 is equation and design analysis and facility development. Phase 3 is coding analysis and testing.

15. End-Product Use

The following documents are called for:

1. A Software Users' Guide is to be provided as part of the AS-BUILT Detailed Software Design Spec
2. A Software Users' Manual
3. A Software Training Plan. This plan supports the need to provide software checkout and mission support personnel the technical knowledge necessary to perform their work.

16. Maintenance

Covered in item 6, O&M Plan

C. Specialized Needs

17. Data Collection

Book 2, Software Configuration Control, includes and describes the following required forms:

1. Discrepancy Record (DR). This record covers design implementation errors or deficiencies subsequent to document approval.
2. Test Discrepancy Record (TDR). This record covers design implementation errors or deficiencies recognized during testing.
3. Software Problem Report (SPR). This record covers software problems.

## 18. Procurement

The description of the conceptual phase of the software life cycle in Book 1 states that the MSFC Software organization is required to review the projects approach to the contracting of software development and management and to participate in source evaluation and contract negotiation.

Book 3, Documentation Deliverables and Planning Data, outlines typical software deliverables.

## Certification

See Validation and Q&A.

### A-1.3 GSFC/Landsat-D Characterization

The Landsat-D program was chosen as the focus of attention for this study of GSFC software management practices. This experimental characterization does not address the flight software produced internal to GSFC nor the ground flight-support software produced internally by GSFC and/or "captive" contract personnel. The documentation, produced by General Electric with the approval of GSFC, is felt to indicate policies which are representative of GSFC flight software practices. It is recognized that the GE SEAM document on which Landsat-D contractor planning is based, is written to be compatible with the corporate structure of General Electric and the services that are supplied by the elements of that structure. Both the GE SEAM and GE generated Landsat-D software management documentation were reviewed.

The policies and procedures contained within the SEAM are conventional and represent the recent state of the evolution of software (and systems in general) management. Documentation of policies and procedures in this fashion provides managers a "check list" to expedite the planning and implementation of policies, procedures and documents suitable for their projects. There is a substantial level of detail in the SEAM which would be of use to projects of all sizes, particularly those small in size or low in resources, in that it provides documents, formats and verbiage which can be used directly or with varying degrees of modification and deletion.

A review of documentation supplied to CSDL in support of the Landsat-D project peer-group review and the recommendations and comments of the reviewing body provide insight into the relationship of the General Electric SEAM to the GSFC Landsat-D software policies. It is not clear; however, what the GSFC Landsat-D software management policies are, in the more general sense, or how various responsibilities are divided between GSFC and GE.

## GENERAL COMMENTS

The GE SEAM (Software Engineering And Management) document delineates the software "approach" of the GE Space Division. The document has four distinct levels of detail: (1) policy, (2) general instructions, (3) standards and practices and (4) guidelines. The policy consists of mandatory and recommended practices.

The SEAM document content at the four levels of detail are:

- 1) Policy - details the provisions of SEAM.
- 2) General Instructions - mandate software engineering and management policies using a top-level representation of the process model and a brief elaboration of the items in the model. The intent, key relationships and provisions of each plan are provided.
- 3) Standard Outline - provides skeleton outlines defining content of all documents supportive of mandated SEAM policies.
- 4) Guidelines - examples, annotated outlines and guides to support 2) and 3) above.

The Standard Outlines, with the exception of the Software Management plan, and the Guidelines were not included in the SEAM document reviewed.

The document is well organized and easily read and understood. There is a consistent level of detail. However, on occasion detail is made available at a level that should have been presented at a higher level (e.g. a list of mandatory vs. recommended documentation shows up first in the standard outline section when it would have been more appropriate in the general instruction section).

Some information in the SEAM is "hidden". For example -- references to a "Unit Test Plan" are made in the sections on "Software Test - Program Plan" and "Methodology Directive" but its contents isn't described in either.

In general the using project is free to tailor the SEAM mandated policies to their needs as long as the intent of the policies are met.

## COMMENTS ON REVIEWED DOCUMENT

### I. ITEMS NECESSARY FOR MANAGEMENT

#### Top Level Policy

- SEAM manual has section on "General Policy". The section covers SEAM purpose, program policy and individual responsibilities.
- Top-level discussion of policy and objectives

#### Program Management Plan

- Software Project Management Plan - A general instruction outline and a document skeleton outline are included. (GI-1).
- Content of Document is specified
  - 1) Project Objectives
  - 2) Organizational Responsibilities
  - 3) Management Methodology
  - 4) Engineering Methodology
  - 5) Baselines
  - 6) Test Concept
  - 7) Configuration Management Concept
  - 8) Quality Assurance Concept
  - 9) End-product Turnover
  - 10) Software Development-work Structure
  - 11) Documentation

Guidelines and cross relationships to other sections of the SEAM document are given.

- Standard outline provides detailed information on the required contents of the plan.

- Recognizes that for embedded computer systems, software management is a prime element of hardware-software system management.
- Each project tailors SEAM plan to fit.
- The Landsat-D Software Management plans follow the SEAM closely with the exception that end-product turnover is not addressed as in the SEAM.

### Configuration Control

- Software Configuration Management Plan - a general instruction outline, a general instruction and a document skeleton outline are included.
- Content of document is specified
  - 1) Organizational Responsibilities
  - 2) Configuration Items Identification
  - 3) Configuration Control
  - 4) Configuration Status Accounting
  - 5) Verification Implementation
- Timing of plan is specified.
- No guide as to "what" should be controlled is supplied in the SEAM; however the Landsat-D Software Management plan does define baseline configuration management applicability.
- Landsat-D document defines methodology used in project.
- The intent of the plan and key relationships are defined.

### Quality Assurance

- The SEAM document delineates the intent, key relationships and provisions of the Quality Assurance Plan.
- The SEAM document does not provide "how-to" information or the details of "what-to". Details of this sort are left to project/customer.
- The General Instruction, is about the right level for a Center level management instruction.

The Landsat-D documentation defines the QA function as follows.

- 1) QA auditor must be independent of technical management being audited.
- 2) QA auditor responsible for implementation of QA plan.
- 3) QA auditor reviews and audits software documentation and the software product during the life cycle.
- 4) QA participates in:
  - a) Configuration control;
  - b) Testing activities including plans, reports and procedures. Assures only controlled items used in test and that test data records are collected;
  - c) Participates in appropriate reviews and audits;
  - d) Inspection of documentation; and
  - e) Provides instruction for handling, storage, etc. of all program software products.
- 5) The degree of QA involvement is not defined.

#### End-Product Acceptance

- Software end-product turnover plan identifies the end-products and services to be presented to the customers; and prescribes the times, conditions and documentation supporting delivery to and acceptance by the customer.
- The SEAM document delineates the intent, key relationships and provisions of the end-product turnover plan.
- Implementation of turn-over is addressed in the SEAM software project management plan and the software program test plan although product acceptance is not specifically covered in the SEAM sections of those plans.
- The Landsat-D software management plan does not specifically address end-product acceptance.

### Operations and Maintenance Plan

- This topic is not covered by the SEAM document at the General Instruction level although a document outline is provided.

The Landsat-D Software Management plan does not address the operation and maintenance phase.

### Resource and Schedule Estimation, Monitoring and Control

- This topic is not covered by the SEAM document at the General Instruction or higher level although a document outline, not available for review, is specified.
- The Landsat-D Software Management plan does not address the subject.

### Risk Assessment

- This topic is not covered by the SEAM document at the General Instruction, or higher, level although a document outline, not available for review, is specified.
- The Landsat-D documentation does not address the subject.

### Documentation

- The SEAM software project management plan lists all policy mandated and suggested documentation. Outlines and guidelines are provided for all policy mandated documentation.
- Documentation such as "Version Description Documents", "Hardware/Software Interface Specification" and "Software Development Plan" are not SEAM policy mandated but are left to the discretion of project management.



- The Landsat-D Software Management plan lists as required:

- a) Software Management Plan
- b) Software Subsystem Requirements Specification
- c) Computer Program Design Specification
- d) Users Manual
- e) Test Plan
- f) Interface Control Document
- g) Programming Practices, Standards and Conventions
- h) Quality Assurance Plan
- i) Configuration Control Plan

## II. ITEMS NECESSARY FOR DEVELOPMENT

### Development Plan

There is no policy mandate nor are there guidelines for a development plan; however, the project management plan, as outlined in the SEAM, covers many of the aspects of a development plan. Development plan is listed as a project option. The following development milestones are specified in the Program Management Plan.

- Three review milestones are identified in SEAM; requirements, preliminary design and critical design.
- The SEAM specifies the intent of the review, prerequisites for the review and conduct of the review as well as responsibilities of the project manager; brief and to the point.
- The Landsat-D formal, GSFC witnessed, reviews, as specified in the software management document, are requirements review, conceptual design review and detailed design review. Other informal and internal reviews include design walk throughs and code reading performed by peers.

## Requirements Specification

- Two types of requirements specification are used; segment software (GI-6) and software (GI-7).
- Methodology for requirements generation is not covered.
- The SEAM specifies intent, key relationships, content and use in the General Instruction section. The level of detail is perhaps appropriate to an NMI. While the content is generally specified, the details of implementation are left to be defined by the project.
- The Table of Contents lists a section for the software requirements document guidelines which was not included in the reviewed document.
- Outlines are provided for both segment software and subsystem software requirements.

## Design Specifications

- Covers "Build to" (prior to CDR) and "as Built" (prior to delivery).
- The SEAM specifies intent, key relationships, content and use. The content section is especially well written; providing, for example, the details that will be given in the specification for each software component.
- The Landsat-D documentation specifies two design specifications:
  - 1) Computer program design specification (preliminary)
  - 2) Computer program design specification

The preliminary version consists of overall structure and functions described in top level terms. Definition of the interfaces between each software unit and methods of data transfer are also included.

The Final Design Specification is a completed version of the "preliminary" and at the time of baselining will be complete and in "code-to" detail. The specification includes detail process flows, data organizations, etc. Design walk throughs are used to detect misunderstandings and deficiencies.

## Program Development

- Program development in the general sense is not addressed by the SEAM. However, two related aspects are covered.

- 1) Software Engineering Methodology - provisions supplied are:

Design practices

Walk throughs, UDFs

Implementation tools

Test standards

Internal Change Control Procedures (DRs, etc.)

Programming Practices

Program Libraries

- 2) Programming Practices, Standards and Conventions invoked by Software Engineering Methodology (see above) - Contents include:

Documentation Standards - Flow charts, Control and data flow, etc.

Coding standards

Data structures

Conventions

## Test Evaluation and Demonstration

- Software Program Test Plan (SEAM)

- 1) Intent:

1. Strategy and approach to formal testing
2. Ensure testing compatible with implementation
3. Identify resources and facilities
4. Organizational responsibilities

- 2) Relationships:

1. To system and segment test plans
2. Between formal and informal development test
3. Discrepancy procedures, comments, QA and engineering methodology directive (GI-13)

**3) Provisions:**

- 1. Organizational Responsibilities**
- 2. Test Phases and Levels**
- 3. Facilities and Resources**
- 4. Data Bases**
- 5. Test Software**
- 6. Test Management Procedures**
- 7. Traceability Audit Support**
- 8. Test Case Specification**
- 9. Reviews**

- SEAM Standards/Unit Test Plan is covered in the Software Engineering Methodology Directive. The content of a test standard is provided in brief terms with the specifics to be provided by the project.
- Verification and Validation are covered in SEAM sections S-15 and GG-15 which were not in the material available for review.
- Guidelines for the software test program plan are listed in the Table of Contents but were not in the document reviewed.
- The standard outline for the "Software Test - Program Plan" provides outlines for the plan, test procedures and test reports. The outlines define purpose and intent, detail left to project. The plan outline contains:
  - 1) Introduction - scope, objectives...
  - 2) Subsystem Definition - identification, summary
  - 3) Verification Philosophy - methods and approach facilities...
  - 4) Scheduling - milestones, schedules, risk areas
  - 5) Test Specifications - methods, equipment...
  - 6) Evaluation - test analysis, evaluation techniques
  - 7) Test Management - control of tests, reviews, data base...
  - 8) Traceability - requirements

## End-Product Use

- The SEAM coverage of end-product use is limited to "S/W Users Manuals" and an "Operation and Maintenance Plan".
- S/W Users Manual - Intent is to provide a complete Description of how to use the software end-product. The content of the document is:
  - 1) Description (overall)
  - 2) Assembly - how to compile, link, etc.
  - 3) Operation
  - 4) Inputs/outputs description
  - 5) Error recovery

An outline is provided but each project is responsible for details.

- The "Operation and Maintenance Plan" was discussed in Category I.
- A standard outline for the "Software Users Manual" is provided.

## Maintenance

- Discussed as "Operations and Maintenance"

## III. SPECIALIZED NEEDS

### Data Collection

- Statistics report (B-6, policy) - "Data concerning the effectiveness of each project's software engineering and management processes shall be collected and analyzed. Steps shall be identified for improvement, particularly in the areas of cost and quality".

The Landsat-D software Management plan does not address data collection.

### Procurement

- Not covered by SEAM.

### Certification

- Certification is not addressed by SEAM. However outlines and guidelines exist for verification and validation (not reviewed). For NASA-type programs, certification can be thought of as an extension of V&V.

A-1.4 Characterization of JPL/Galileo  
Software Management



JET PROPULSION LABORATORY

PROJECT GALILEO

- GALILEO SOFTWARE MANAGEMENT PLAN, VOLUME 1, TECHNICAL DEVELOPMENT AND MANAGEMENT POLICIES AND APPROACH 625-510, REV B. OCTOBER 18, 1979
- VOLUME 11, SOFTWARE TECHNICAL AND ADMINISTRATIVE PROCEDURES (STAPS)
- COMMENTS FOLLOW IMMEDIATELY

STANDARD PRACTICE

- STANDARD PRACTICES FOR THE IMPLEMENTATION OF COMPUTER SOFTWARE A.P. IRVINE, EDITOR  
JPL PUBLICATION 78-53, SEPTEMBER 1, 1978
- JPL STANDARD PRACTICE, IMPLEMENTATION AND MANAGEMENT OF JPL SOFTWARE DRAFT-APRIL 15, 1980, PRELIMINARY
- COMMENTS FOLLOW AFTER GALILEO

## SECTION 1 INTRODUCTION

- THIS PLAN CONTAINS METHODS, DESIGN & DEVELOPMENT PROCESS, PROCEDURES, STANDARDS, DOCUMENTATION REQUIREMENTS AND SOFTWARE DESIGN PHILOSOPHY
- PLAN COVERS FULL LIFE CYCLE
- PLAN IS FULLY RESPONSIVE TO NASA NMI 2410.6
- DETAILS ARE COVERED IN VOL II (QUITE THICK, NOT REVIEWED) VIA 'STAPS' (SOFTWARE TECHNICAL AND ADMINISTRATIVE PROCEDURES). 'STAPS' ARE CUMULATED IN MEMORANDUM FORMAT.



### REGRESSION TEST BASELINE

ONE OR TWO SMALL TEST CASES FOR EACH PROGRAM USED PREVIOUSLY  
IN ACCEPTANCE TESTING TO:

- 1) CHECK MODIFIED SOFTWARE
- 2) CHECK PROPOSED SYSTEM LEVEL CHANGES

SOFTWARE IS

- 1) COMPUTER INSTRUCTIONS & DATA INCLUDING MICRO CODE
- 2) DOCUMENTATION OF ABOVE
- 3) OBJECT CODE OF ABOVE

- GALILEO WILL INHERIT MUCH VOYAGER SOFTWARE,
- GALILEO SHALL USE HAL/S (OR HAL/X?) AS ITS HIGHER ORDER LANGUAGE
- PDL USED FOR DESIGN PHASE
- FLIGHT SOFTWARE REPROGRAMMABILITY COVERED
- MARGIN MANAGEMENT COVERED
- SOFTWARE SYSTEM ENGINEERING DESCRIBED IN DETAIL IN SECTION 1.8





- SOFTWARE LIFE CYCLE
  - REQUIREMENTS DEVELOPMENT
  - PRELIMINARY DESIGN
  - DETAILED DESIGN
  - CODING AND DEBUGGING
  - TESTING
  - INTEGRATION AND DELIVERY
  - MAINTENANCE
- PLAN IS SPECIFICALLY ORGANIZED TO BE RECYCLED AS A GENERAL JPL PLAN (SECTIONS 2-4) WITH SECTION 5 REWRITTEN AS THE PROJECT SPECIFIC EXCEPTION LIST.
- FIGURE 1-1 SHOWS THE "UMBRELLA" OF THE JPL INSTITUTION ENVIRONMENT OVER THE GALILEO PLAN. UMBRELLA INCLUDES NASA NMI 2410.6, DSN, MCCC, HISTORY.
- SOFTWARE THINKING GROUP - AD HOC GROUP TO DISCUSS DESIGN ISSUES. MINUTES ARE KEPT BY THE SSE STAFF.
- THE PLAN REQUIRES THAT PORTABILITY BE ADDRESSED, EVEN IF IT CANNOT BE ACHIEVED.



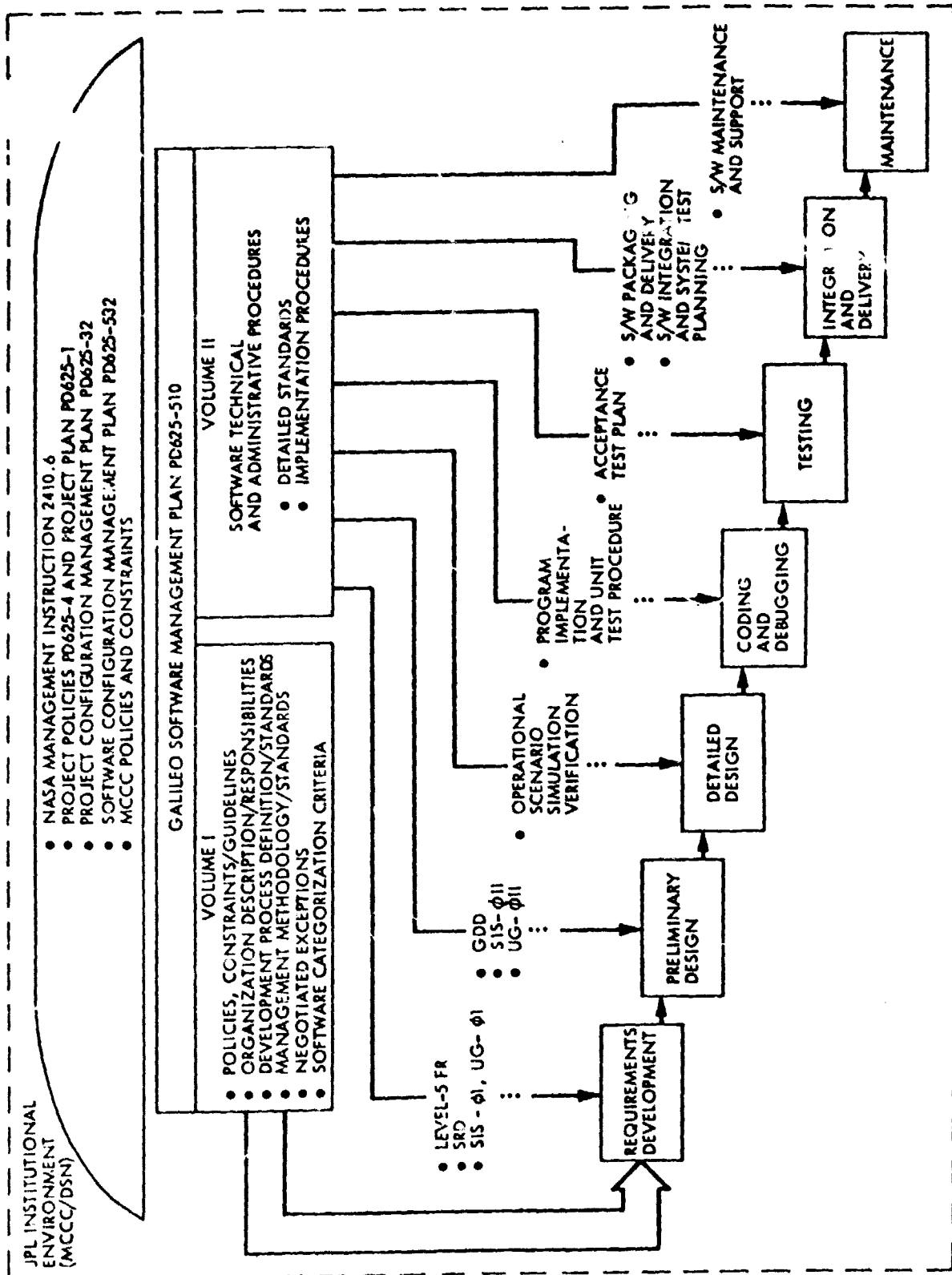


Figure 1-1. Plan Organization Overview

## SECTION 2 ORGANIZATION

- "TYPICAL PROJECT/LINE MATRIX STRUCTURE";
- PROJECT OFFICE IS PARALLEL DIVISION TO THE TECHNICAL DIVISIONS,
- SOFTWARE SYSTEM MANAGER (SSM)-FROM THE PROJECT OFFICE,  
RESPONSIBLE TO THE MISSION OPERATIONS AND ENGINEERING MANAGER,
- STAFF FOR THE SSM - FROM THE TECHNICAL DIVISIONS
  - GROUND SOFTWARE SYSTEM ENGINEER GSSE
  - SPACECRAFT SOFTWARE SYSTEM ENGINEER S/C SSE
  - INCLUDES SCIENCE INSTRUMENT  
SOFTWARE ENGINEER (SISE)
  - INDEPENDENT V&V CONTRACTOR - MARTIN MARIETTA
- EACH DIVISION HAS A PROJECT REPRESENTATIVE WHO HAS AUTHORITY TO DECIDE  
ALL DIVISION RELATED MATTERS FOR THE PROJECT.
- EACH DIVISION SUPPLIES AN ASSISTANT PROJECT REPRESENTATIVE FOR SOFTWARE  
WHO SERVES DUALY ON THE SOFTWARE SYSTEM ENGINEERING STAFF.
- EACH DIVISION HAS A TYPICAL ORGANIZATION OF DIVISION LINE MANAGER,  
SECTION MANAGERS, AND GROUP SUPERVISORS.

### SECTION 3 SOFTWARE DEVELOPMENT PROCESS

- THE 42 PAGES OF SECTION 3 COVERS THE SOFTWARE LIFECYCLE, WITH EACH 13 MILESTONES, ITS DOCUMENTS, REVISIONS, AND CONTROLS
- ONE OUTSTANDING FEATURE IN THE LIFECYCLE DEFINITION IS THE DISTINCTION THAT AFTER THE COGNIZANT ENGINEER HAS COMPLETED THE FINAL DETAILED REQUIREMENTS DEFINITION, IN PARALLEL HE GENERATES THE ACCEPTANCE TEST PLAN FORMULATION WHILE THE COGNIZANT PROGRAMMER GENERATES THE DETAILED DESIGN DEFINITION FOR THE PDR.
- THE LIFECYCLE STEPS ARE INTENDED FOR "INHERITED SOFTWARE" AS WELL AS DEVELOPED SOFTWARE. THIS SEEMS AMBITIOUS.
- TABLE 3-1 LISTS, MILESTONES, DOCUMENTS, RESPONSIBILITIES, REVIEW TEAMS, APPROVALS, AND CHANGE CONTROL

## SOFTWARE LIFECYCLE STEPS

- (1) PROJECT SYSTEM FUNCTIONAL REQUIREMENTS AND DESIGN SPECIFICATIONS DEFINITION (3.2).
- (2) SUBSYSTEM FUNCTIONAL SPECIFICATIONS DEFINITION AND SOFTWARE FUNCTIONAL DESCRIPTION (3.3).
- (3) INTEGRATED SOFTWARE FUNCTIONAL DIAGRAM (3.4).
- (4) DETAILED SOFTWARE REQUIREMENTS DEFINITION (3.5).
- (5) OPERATIONS AND USER INPUT SPECIFICATIONS (USER'S GUIDE) DEFINITION (3.6).
- (6) SOFTWARE INTERFACE SPECIFICATIONS DEFINITION (3.7).
- (7) ACCEPTANCE TEST PLAN FORMULATION (3.8).
- (8) PROGRAM GENERAL DESIGN DEFINITION (3.9).
- (9) PROGRAM DETAILED DESIGN AND RESOURCE ESTIMATE AND SCHEDULE DEFINITION (3.10)
- (10) PROGRAM OPERATIONS SCENARIO SIMULATION VERIFICATION (3.11).



SOFTWARE LIFECYCLE STEPS (CONTINUED)

- (11) PROGRAM IMPLEMENTATION AND UNIT TESTING (3.12).
- (12) ACCEPTANCE TESTING (3.13).
- (13) SOFTWARE INTEGRATION AND SYSTEM TEST PLAN FORMULATION (3.14).
- (14) PROGRAM PACKAGING AND DELIVERY (3.15).
- (15) SOFTWARE INTEGRATION AND SYSTEM TESTING (3.16).
- (16) SOFTWARE MAINTENANCE (3.17).
- (17) SOFTWARE POST-LAUNCH DEVELOPMENT (3.18).

#### SECTION 4 SOFTWARE MANAGEMENT METHODOLOGY

- GOAL OF THIS SECTION IS TO DETAIL WHAT MEASUREMENT TECHNIQUES ARE AVAILABLE TO THE MANAGER TO MAKE INTELLIGENT TRADEOFFS BETWEEN REQUIREMENTS AND BUDGET/SCHEDULE RESOURCES.
- SEVERAL LAYERS OF CHANGE CONTROL AUTHORITY ARE SPECIFIED SO THAT TIMELY DECISIONS ARE POSSIBLE.
- INCLUDES THE REGULARLY SCHEDULED MEETINGS
  - PROJECT MONTHLY MANAGEMENT REVIEW
  - PROJECT STAFF MEETINGS
  - PROJECT QUARTERLY REVIEW
  - SRM UPDATES (SYSTEM FOR RESOURCE MANAGEMENT)
  - SEMI-ANNUAL BUDGET EXERCISE
  - RESOURCES STATUS (RSR), MONTHLY
- THE IV&V CONTRACTOR, MARTIN MARIETTA, PROVIDES, IN ADDITION, A SOFTWARE AUDIT.
- A VERY BRIEF REFERENCE IS MADE TO CATEGORIZING SOFTWARE - BETWEEN A) MISSION CRITICAL AND B) OTHER.
- EACH TECHNICAL DIVISION OF JPL GENERATES ITS OWN SOFTWARE IMPLEMENTATION PLANS (DSIP)

## SOFTWARE CONFIGURATION MANAGEMENT

- COVERED IN A SEPARATE DOCUMENT (PD 625-532, NOT REVIEWED), BUT SUMMARIZED IN THIS PLAN.
- FOUR PRODUCT BASELINES ARE DEFINED
  - REQUIREMENTS (BUILD-TO)
  - DESIGN (CODE-TO)
  - PROGRAM (AS-BUILT)
  - REGRESSION BASELINE (AS-BUILT COMPLIANCE)
- CHANGE CONTROL MAY BE LOCALLY CONTROLLED FOR NON-MISSION CRITICAL SOFTWARE, BUT MUST BE FORMALLY CONTROLLED BY THE SSE IF MISSION CRITICAL.
- AN EMERGENCY PROCEDURE EXISTS FOR URGENT NEEDS.



SOFTWARE VALIDATION, VERIFICATION, AND QUALITY ASSURANCE.

- INTERNAL AUDIT AND VERIFICATION WILL INCLUDE PEER REVIEWS BY THE TECHNICAL DIVISIONS
- EXTERNAL AUDIT AND VERIFICATION SEEKS REVIEW BY PERSONS NOT "BIASED TO THE SOFTWARE" IN THREE WAYS
  - VIA THE PROJECT REVIEWS WITH USERS AND NON JPL PERSONNEL PRESENT
  - VIA AN IV&V CONTRACTOR (MARTIN MARIETTA)
  - VIA INDEPENDENT CONSULTANTS
- QUALITY ASSURANCE WILL BE CONDUCTED BY JPL QA&R DIVISION FOR
  - CONFORMANCE TO STANDARDS
  - TRACEABILITY.
- QA AUDITS WILL BE PERFORMED ON TYPICALLY 10% OF THE CATEGORY A CODE, AND CERTAIN SOFTWARE DEFINED AS CRITICAL WILL BE AUDITED IN FULL.

## REQUIREMENTS DEFINITION

- THE IMPORTANCE OF "COMPLETE, UNAMBIGUOUS, AND WELL DEVELOPED REQUIREMENTS" IN THE FORM OF "EQUATIONS AND ALGORITHMS THAT ARE VALIDATED AND COMPLETE (MATURE)" IS STRESSED.
- WHERE NOT COMPLETE, DEFICIENCIES ARE TO BE NOTED IN THE REQUIREMENTS DOCUMENT AND THEN TRADED BY THE SCR SYSTEM AFTER INITIAL APPROVAL.

## DESIGN APPROACH AND METHODOLOGY

- THE SOFTWARE FUNCTIONAL DESCRIPTION (SFD) AND INTEGRATED SOFTWARE FUNCTIONAL DESCRIPTION (ISFD)  
CALL FOR - TOP DOWN DEVELOPMENT
  - FORMAL STRUCTURED DESIGN LANGUAGE

#### DESIGN REVIEWS (PDR AND CDR)

- THE PRELIMINARY DESIGN REVIEW (PDR) INCLUDES A PEER GROUP WALKTHROUGH OF THE GENERAL DESIGN DOCUMENT.
- A REEVALUATION OF SCHEDULE AND COSTS FOLLOWS THE PDR
- THE CRITICAL DESIGN REVIEW IS SCHEDULED FOR THE TIME THAT 50% OF THE MODULES ARE COMPLETED.
- A CHECKLIST FOR EACH DESIGN REVIEW IS INCLUDED.

## SOFTWARE TESTING

- IN ADDITION TO THE UNIT TESTING THAT ACCOMPANIES CODING, THREE LEVELS OF TESTING ARE SPECIFIED:
  - TESTING AS A STAND ALONE PRODUCT BY THE CE
  - TESTING AS A SUB-SYSTEM - UNDER AUSPICES OF THE SOFTWARE SYSTEM ENGINEER USING THE IV&V TEAM AND OTHERS
  - TESTING AS A FULL SYSTEM - END TO END SCENARIOS ARE EXECUTED IN A PSEUDO-REALISTIC OPERATIONAL ENVIRONMENT



PROJECT GALILEO, SOFTWARE MANAGEMENT PLAN  
 VOL I TECHNICAL DEVELOPMENT AND MANAGEMENT  
 POLICIES AND APPROACH  
 625-510, REV. B

- APPENDIX A DESCRIBES IN DETAIL THE DUTIES OF EACH MEMBER OF THE SOFTWARE TEAM: NAMELY  
 SOFTWARE SYSTEM MANAGER (SSM)  
 GROUND SOFTWARE SYSTEM ENGINEER (GSSE)  
 SPACECRAFT SOFTWARE SYSTEM ENGINEER (S/C SSE)  
 SCIENCE INSTRUMENT SOFTWARE ENGINEER (SISE)  
 ASSISTANT PROJECT REPRESENTATIVE FOR SOFTWARE (APRS)  
 TECHNICAL DIVISION SYSTEM ENGINEER (SE)  
 INDIVIDUAL PROGRAM COGNIZANT ENGINEERS (CE)  
 COGNIZANT PROGRAMMER (SP)

- APPENDIX B LISTS ALL SOFTWARE AFFECTING GALILEO. THIS INCLUDES SPECIFIC FLIGHT SOFTWARE (IN ONE OF 19 RCA 1802 COMPUTERS OR 2 ATAC 16's), AS WELL AS MISSION SUPPORT AND NON-REAL TIME GENERAL PURPOSE SOFTWARE AT JPL NOT GALILEO SPECIFIC. IT ALSO INCLUDES THE DEEP SPACE NETWORK (DSN) SOFTWARE



- APPENDIX C LISTS THE STEPS TO REQUEST  
DEVIATION FROM THIS PLAN. IT DOESN'T  
INCLUDE AN EXAMPLE.
- APPENDIX D DEFINES 2 CATEGORIES OF  
SOFTWARE
  - A - MISSION CRITICAL - EXEMPT FROM  
CONTROLS ONLY BY SOFTWARE MANAGER
  - B - OTHER - RESPONSIBILITY OF  
ORGANIZATIONS REQUIRING IT



## STANDARD PRACTICE

- THE SECOND STANDARD PRACTICE DOCUMENT IS AN IN PROGRESS UPDATE TO THE 1978 EDITION, SEEN IN PRELIMINARY FORM ONLY
- THE MAJOR FEATURES OF THE UPDATE
  - SIMPLIFICATION
  - DEFINITIONS OF DIFFERENT POLICIES AND PRACTICES FOR DIFFERENT CLASSES OF SOFTWARE
  - ADDRESSING THE VARIOUS RETURN PATHS TO PRECEDING PHASES IN ORDER TO ACCOMMODATE ALTERATIONS IN REQUIREMENTS, DESIGN CHANGES, AND CODE CORRECTION
- STANDARD PRACTICES IS COMPLETE AS A STAND ALONE GUIDE TO SOFTWARE DEVELOPMENT, INCLUDING LIFE CYCLE DESCRIPTION (SLIGHTLY DIFFERENT FROM GALILEO, SO NOT INCLUDED HERE), MILESTONES, REVIEWS
- IN ADDITION, IT INCLUDES THE DETAILED HOW TO FOR MANY STEPS, INCLUDING FORMS, AND EXAMPLES





This section (A-1.5) is proprietary  
to TRW and requires their  
permission before distribution

A-1.5 Characterization of TRW  
Software Management



TRW POLICY AND STANDARDS DOCUMENTS  
FOR SOFTWARE MANAGEMENT

DOCUMENTS REVIEWED

- SYSTEMS ENGINEERING AND INTEGRATION DIVISION  
SOFTWARE DEVELOPMENT POLICIES OF 1/3/78 (WITH  
EACH OF THE EIGHTEEN ASSOCIATED POLICY STATEMENTS  
INDIVIDUALLY DATED)
- SYSTEMS ENGINEERING AND INTEGRATION DIVISION  
SOFTWARE PRODUCT STANDARDS, 00145, FEBRUARY 28, 1977

## SEID POLICIES

- THE SEID POLICIES COVER ALL SOFTWARE PROJECTS OF A SPECIFIED SCOPE, DURATION, NUMBER OF COMPUTER PROGRAMS, AND NUMBER OF USERS
- A PERMANENT BOARD IS ESTABLISHED TO CONTROL AND MAINTAIN THE POLICIES (TRAFFIC BEFORE THE BOARD WAS INITIALLY HEAVY THEN BECAME VERY LIGHT)
- DEPARTURE FROM POLICIES REQUIRES A WRITTEN AUTHORITY
  - A DEVIATION IS REQUESTED IN ADVANCE
  - A WAIVER IS REQUESTED AFTER ACTIVITY IS INITIATED

● THE EIGHTEEN SPECIFIC SEID POLICIES ARE

- 1) SOFTWARE REQUIREMENTS SPECIFICATION
- 2) SOFTWARE REQUIREMENTS REVIEW AND ACCEPTANCE
- 3) SOFTWARE DESIGN SPECIFICATION (PRELIMINARY)
- 4) SOFTWARE PRELIMINARY DESIGN REVIEW
- 5) SOFTWARE DESIGN SPECIFICATION (DETAILED)
- 6) SOFTWARE CRITICAL DESIGN REVIEW
- 7) TOP DOWN SOFTWARE DESIGN
- 8) UNIT DEVELOPMENT FOLDERS
- 9) SOFTWARE DESIGN WALK-THROUGHS
- 10) PROGRAMMING STANDARDS
- 11) SOFTWARE UNIT TEST CRITERIA DOCUMENT
- 12) SOFTWARE SYSTEM INTEGRATION AND TEST
- 13) SOFTWARE ACCEPTANCE TEST PLAN AND PROCEDURES
- 14) SOFTWARE USER'S MANUAL
- 15) SOFTWARE END-PRODUCT ACCEPTANCE PLAN
- 16) SOFTWARE DOCUMENTATION
- 17) SOFTWARE CONFIGURATION MANAGEMENT
- 18) SOFTWARE QUALITY ASSURANCE



## 1) SOFTWARE REQUIREMENTS SPECIFICATION

- MUST BE PRODUCED FOR THE SOFTWARE REQUIREMENTS REVIEW (SRR)
- MUST BE SIGNED BY THE CUSTOMER
- MAY ONLY BE CHANGED BY JOINT APPROVAL OF THE CUSTOMER AND TRW
- A DEFAULT FORMAT IS DETAILED IN THE STANDARDS DOCUMENT
- CONTAINS
  - FUNCTIONAL REQUIREMENTS
  - PERFORMANCE REQUIREMENTS
  - INTERFACE REQUIREMENTS
- MANDATORY REQUIREMENTS ARE DISTINGUISHED FROM GOALS OR OPTIONS



## 2) SOFTWARE REQUIREMENTS REVIEW (SRR) AND ACCEPTANCE

- PURPOSE OF SRR IS TO ACQUIRE CUSTOMER AGREEMENT FOR THE REQUIREMENTS SPEC, WHICH IS THE BASIS FOR END PRODUCT ACCEPTANCE
- PREPARATION FOR THE SRR SHALL INCLUDE ANALYSIS FOR  
CLARITY  
COMPATIBILITY WITH SYSTEM LEVEL OBJECTIVES  
TECHNICAL FEASIBILITY  
TESTABILITY  
COMPLETENESS
- SCHEDULE, COST AND AVAILABILITY OF NECESSARY RESOURCES SHOULD BE REVIEWED
- UNRESOLVED ISSUES ARE PRESENTED AT THE SRR
- IDENTIFIED PROBLEMS ARE PART OF THE WRITTEN RESULTS

### 3) PRELIMINARY SOFTWARE DESIGN SPECIFICATION

- MUST BE SUFFICIENT TO GENERATE THE DETAILED DESIGN SPEC
- A DEFAULT FORMAT IS AVAILABLE IN THE STANDARDS DOCUMENT
- MUST ASSIGN EACH UNIQUE REQUIREMENT TO A SOFTWARE COMPONENT
- MUST SPECIFY AND NAME ALL LEVELS OF SOFTWARE, WITH
  - FUNCTIONAL DESCRIPTION
  - CONTROL AND DATA INTERFACES
  - DATA PROCESSING FLOW
  - MODIFICATION APPROACH IF EXISTING SOFTWARE TO BE USED
- INCLUDES
  - DATA PROCESSING RESOURCE BUDGETS
  - DATA BASE HIERARCHY TO A SPECIFIED LEVEL OF DETAIL
  - USERS INTERFACE

- 4) SOFTWARE PRELIMINARY DESIGN REVIEW (PDR)
  - PURPOSE OF THE PDR IS TO REVIEW ADEQUACY OF WORK TO DATE AND GET AGREEMENT TO PROCEED TO THE DETAILED DESIGN PHASE FROM
    - THE PROJECT REVIEW AUTHORITY (APPOINTED BY THE PROJECT MANAGER)
    - THE PROJECT MANAGER
    - THE CUSTOMER
  - THE REVIEWED MATERIAL IS
    - A) THE PRELIMINARY DESIGN SPECIFICATION
    - B) THE APPROVED SOFTWARE REQUIREMENTS SPECIFICATION
    - C) PROPOSED CHANGES TO THE SOFTWARE REQUIREMENTS SPECIFICATION
    - D) A PRELIMINARY USERS MANUAL
    - E) A PRELIMINARY ACCEPTANCE TEST PLAN
    - F) PRELIMINARY PERFORMANCE ESTIMATES
  - THE PROJECT TEAM MUST COMPLETE ITS OWN VERIFICATION OF THE REVIEWED MATERIAL BEFORE PDR
  - CONTRACTURAL ISSUES AND AGREEMENTS CO-SIGNED BY TRW AND THE CUSTOMER ARE AN OUTPUT



5) DETAILED SOFTWARE DESIGN SPECIFICATION

- THE PRELIMINARY SPEC IS UPDATED TO A "BUILD-TO-" BASELINE
- INCLUDES
  - AN OVERVIEW OF THE OVERALL COMPUTER
  - TIMING, STORAGE, AND ACCURACY MARGINS
  - DETAILED DESCRIPTION OF THE DATA BASE TO THE BIT LEVEL
  - FOR EACH ROUTINE
    - A. ROUTINE NAME
    - B. PURPOSE (BRIEF NARRATIVE SUMMARY)
    - C. ASSUMPTIONS
    - D. SIZING (CODE AND DATA)
    - E. CALLING SEQUENCE, ARGUMENTS AND DEFINITIONS, AND ERROR EXITS
    - F. INPUTS, OUTPUTS, AND USE
    - G. ROUTINES CALLED BY THIS ROUTINE, AND CALLING ROUTINES
    - H. ENGINEERING DESCRIPTION
    - I. RESTRICTIONS AND LIMITATIONS
- AFTER INTEGRATION TESTING, THE DESIGN SPEC IS UPDATED TO THE "AS-BUILT" BASELINE

6) SOFTWARE CRITICAL DESIGN REVIEW (CDR)

- PURPOSE OF THE CDR IS TO REVIEW THE ADEQUACY OF THE WORK TO DATE AND GET AGREEMENT TO PROCEED TO THE DEVELOPMENT PHASE FROM
  - THE PROJECT REVIEW AUTHORITY (APPOINTED BY THE PROJECT MANAGER)
  - THE PROJECT MANAGER
  - THE CUSTOMER
- THE REVIEWED MATERIAL
  - A. DETAILED DESIGN SPEC
  - B. CURRENT REQUIREMENTS SPEC WITH INTERFACE SPECS (AND PROPOSED CHANGES)
  - C. CURRENT ACCEPTANCE TEST PLAN
  - D. RESULTS OF DESIGN EVALUATIONS
  - E. UPDATED PERFORMANCE SPECS
- THE PROJECT TEAM MUST COMPILE ITS OWN VERIFICATION OF THE REVIEWED MATERIAL BEFORE CDR
- CONTRACTUAL ISSUES AND AGREEMENTS CO-SIGNED BY TRW AND THE CUSTOMER ARE A WRITTEN OUTPUT OF THE CDR

## 7) TOP DOWN SOFTWARE DESIGN

- TOP DOWN DESIGN MUST BE USED TO ENHANCE
  - DESIGN TRACEABILITY
  - COMPLETENESS
  - COMPREHENSIVENESS
- A DESCRIPTION OF TOP DOWN DESIGN IS INCLUDED IN THE POLICY DOCUMENT
- THE CRITERIA, RATIONALE, AND TRADEOFFS USED ARE TO BE RECORDED IN THE DESIGN SPECIFICATION
- THE POLICY SPECIFICALLY MENTIONS THAT TOP DOWN DESIGN DOES NOT PRECLUDE PROTOTYPING OF CRITICAL LOWER-LEVEL COMPONENTS
- AGAIN THE POLICY ADDRESSES PREEXISTING SOFTWARE THAT IS TO BE MODIFIED

## 8) UNIT DEVELOPMENT FOLDERS

- REQUIRED AS A TRW SEID POLICY TO PROVIDE
  - ORGANIZATION
  - ACCESSIBILITY
- ACCUMULATED AS WORK PROGRESSES
- REVIEWED AS ACCUMULATED BY THE FIRST LEVEL SUPERVISOR FOR
  - TECHNICAL ADEQUACY
  - SCHEDULE COMPLIANCE
- ESTABLISHED WITHIN ONE MONTH OF PDR
- DESCRIBED IN THOROUGH DETAIL IN THE STANDARDS DOCUMENT
- SUBJECT TO QA AUDITS
- CONTAINS (WITH SCHEDULE, STATUS, AND IDENTIFICATION)
  - 1) REQUIREMENTS
  - 2) DESIGN (PRELIMINARY, "CODE-TO", AND "AS-BUILT")
  - 3) FUNCTIONAL CAPABILITIES
  - 4) CODE
  - 5) TEST PLAN
  - 6) TEST CASE RESULTS
  - 7) PROBLEM REPORTS
  - 8) NOTES
  - 9) REVIEWERS COMMENTS



9) SOFTWARE DESIGN WALK-THROUGHS

- CONDUCTED AT THE UNIT LEVEL, AS EACH UNIT IS COMPLETED, BY THE DESIGNER
- PEER LEVEL REVIEW BY ONE TO FOUR PEOPLE
- THE EVENTUAL PROGRAMMER(S) AND TESTER(S) ARE GOOD CANDIDATES
- ITEMS REVIEWED
  - RESPONSIVENESS TO THE REQUIREMENTS
  - COMPLETENESS
  - CONSISTENCY
  - FLOW OF DATA THROUGH INPUT/OUTPUT INTERFACES
  - TESTABILITY
  - ERROR RECOVERY PROCEDURES
  - MODULARITY
  - SIMPLICITY
- PROBLEMS DETECTED BY THE REVIEWERS ARE PRESENTED IN WRITING
- POLICIES DO NOT GRANT GO/NOGO AUTHORITY TO THE REVIEWERS

## 10) PROGRAMMING STANDARDS

- EVERY SEID SOFTWARE PROJECT MUST HAVE A STANDARDS DOCUMENT
- THE DEFAULT IS IN THE SEID STANDARDS DOCUMENT
- THE PROJECT REVIEW AUTHORITY MUST APPROVE A PROJECT-SPECIFIC STANDARDS
- COMPLIANCE WITH PROGRAMMING STANDARDS IS SUBJECT TO QA AUDIT
- MINIMUM STANDARDS
  - DESIGN LANGUAGE (WITH DEFINITION OF SYMBOLS AND CONVENTIONS)
  - PROGRAM IMBEDDED COMMENTS (PREFACE TEXT AND IN LINE COMMENTS)
  - CODING STANDARDS FOR EACH LANGUAGE (HOL AND ASSEMBLY)
  - STRUCTURED PROGRAMMING
  - PROJECT UNIQUE CONSIDERATIONS (EG. DUTY CYCLE, MEMORY UTILIZATION, MAXIMUM ROUTINE SIZE.)
- THE SEID STANDARDS DOCUMENT WELL EXCEEDS THE MINIMUM OF THE POLICY



## 11) SOFTWARE UNIT TEST CRITERIA DOCUMENT

- A DEFAULT FORMAT IS AVAILABLE IN THE STANDARDS DOCUMENT
- MINIMUM CONTENTS
  - GENERAL THOROUGHNESS CRITERIA, E.G.
    - COMPUTATIONS WITH NOMINAL, SINGULAR, AND EXTREME DATA
  - ALL INPUT OPTIONS
  - ALL OUTPUT OPTIONS
  - ALL FORMATS
  - ALL ERROR MESSAGES, ALL INFORMATIONAL MESSAGES
  - ALL STATEMENTS
  - ALL BRANCHES
- PROCEDURES TO ASSURE COMPLIANCE, E.G.
  - COMPLIANCE WITH SIZING AND ACCURACY BUDGETS
  - COMPLIANCE WITH INTERFACES REQUIREMENTS
- SUPPORT SOFTWARE REQUIRED FOR UNIT TESTING
- EXTERNAL TEST DATA SUPPLIED (E.G. GFE)



## 12) SOFTWARE SYSTEM INTEGRATION AND TEST

- ACTIVITIES OF THIS PHASE
  - INTEGRATE THE SOFTWARE
  - DEMONSTRATE THE COMPLETE REQUIREMENTS SPEC
  - DEMONSTRATE OPERABILITY OVER A WIDE RANGE
- A DEFAULT FORMAT FOR THE TEST PLAN IS AVAILABLE IN THE STANDARDS DOCUMENT
- PLAN CONTENTS
  - CONCEPTS, GOALS, AND STRATEGY
  - TEST TECHNIQUES FOR EACH ENTITY, AT EACH LEVEL OF INTEGRATION, WITH SOURCE OF TEST DATA
  - WHERE REQUIRED, ORDER OF TESTING
  - STRESS TESTING
  - SUPPORT SOFTWARE FACILITIES, AND ORGANIZATIONS INVOLVED WITH CONSIDERATION OF THEIR AVAILABILITY
- A DISCREPANCY REPORTING SYSTEM SHALL BE ESTABLISHED





13) SOFTWARE ACCEPTANCE TEST PLAN AND PROCEDURES

- A DEFAULT FORMAT IS AVAILABLE IN THE STANDARDS DOCUMENT
- THE ACCEPTANCE TEST PLAN IS A CONTROLLED DOCUMENT, APPROVED BY THE CUSTOMER UNDER WRITTEN AGREEMENT AT THE CDR
- AFTER CDR, MODIFICATIONS ARE APPROVED IN WRITING BETWEEN THE CUSTOMER AND TRW
- THE ACCEPTANCE TEST PROGRAM IS CONDUCTED BY AN ORGANIZATION INDEPENDENT OF THE DEVELOPERS
- CONTENTS OF PRELIMINARY (PDR) VERSION OF THE ACCEPTANCE TEST PLAN
  - RELATION BETWEEN THE TESTING PHASES
  - CRITERIA FOR JUDGING SOFTWARE READY FOR ACCEPTANCE TESTING
  - ACCEPTANCE CRITERIA
  - EACH TEST CASE IN DETAIL
  - SCHEDULE
  - DATA BASE
  - SUPPORT SOFTWARE



**SOFTWARE ACCEPTANCE TEST PLAN AND PROCEDURES (CONTINUED)**

- **ADDITIONAL CONTENTS OF APPROVED (CDR) VERSION OF THE ACCEPTANCE TEST PLAN**
  - **MANDATORY REQUIREMENTS CORRELATED TO SPECIFIC TESTS**
  - **TEST INPUT REQUIREMENTS, OUTPUT REQUIREMENTS, AND ANALYSIS METHOD**
  - **MODIFICATIONS AND RETESTING STRATEGY**
  - **PROTOCOL FOR CUSTOMERS ROLE IN TESTING PHASE**



#### 14) SOFTWARE USER'S MANUAL

- A DEFAULT FORMAT IS AVAILABLE IN THE STANDARDS DOCUMENT
- REVIEWED AT PDR FOR CUSTOMER'S COMMITMENT TO THE MAN / MACHINE INTERFACE
- A PRELIMINARY, BUT COMPLETE, VERSION MUST BE AVAILABLE FOR THE TESTING PHASES
- THE FINAL VERSION IS DELIVERED WITH THE SOFTWARE, REFLECTING THAT "AS-BUILT AND DELIVERED" BASELINE
- MINIMUM CONTENTS
  - HOW TO SET UP, EXECUTE, SELECT OPTIONS, AND INTERPRETE PRINTOUT
  - INSTRUCTION FOR OPERATIONS (WHERE APPROPRIATE)

# 15) SOFTWARE END-PRODUCT ACCEPTANCE PLAN

- A DEFAULT FORMAT IS AVAILABLE IN THE STANDARDS DOCUMENT
- PURPOSE IS TO ACHIEVE, BY AN ORDERLY PLAN, THE CUSTOMER'S ACCEPTANCE OF ALL PRODUCTS AND SERVICES REQUIRED BY THE CONTRACT
- FIRST REVIEWED AT SRR AND APPROVED BEFORE END OF PDR
- CONTENTS:
  - A CHECKLIST OF ITEMS, WITH SCHEDULES, AND CRITERIA FOR DETERMINING READINESS FOR CLOSE-OUT
  - CUSTOMER'S ROLE IN REVIEW AND ACCEPTANCE
- A FILE IS MAINTAINED OF ALL ACCEPTANCE RELATED DOCUMENTS (E.G., RECEIPTS, APPROVALS, TECHNICAL DIRECTIONS LETTERS)
- AN ACCEPTANCE AUDIT IS PERFORMED NEAR CONTRACT END.

## 16) SOFTWARE DOCUMENTATION

- THE STANDARDS DOCUMENT CONTAINS A DEFAULT FORMAT FOR EACH DOCUMENT
- IF THE CONTRACT SPECIFIES ADDITIONAL DOCUMENTS, THE STANDARDS DOCUMENT MUST BE UPGRADED TO INCLUDE ITS FORMAT
- FOR EACH DOCUMENT THE SECTIONS THAT ARE MANDATORY ARE CLEARLY IDENTIFIED
- THE MINIMUM SET
  - REQUIREMENTS SPECIFICATION
  - DESIGN SPECIFICATIONS
  - TEST AND ACCEPTANCE PLANS
  - QUALITY ASSURANCE AND CONFIGURATION MANAGEMENT PLANS
  - USERS MANUAL



## 17) SOFTWARE CONFIGURATION MANAGEMENT

- ACTIVITIES:
  - ESTABLISH A SERIES OF BASELINES
  - ESTABLISH METHODS TO CONTROL CHANGES
- DEGREE OF FORMALITY AND CONTROL SHOULD REFLECT
  - MANPOWER USED
  - SIZE AND COMPLEXITY
  - SIGNIFICANCE
  - INVESTMENT RISKS
- A PLAN IS REQUIRED - THE DEFAULT IS AVAILABLE IN THE STANDARDS DOCUMENT
- MASTER COPIES ARE CONTROLLED IN A PRODUCT DEVELOPMENT LIBRARY
- PLAN CONTENTS
  - BASELINES USED (MINIMUM OF REQUIREMENTS, DESIGN, TEST, AND PRODUCT)
  - BASELINE DEFINITIONS, REVIEW AND APPROVAL EVENTS

## SOFTWARE CONFIGURATION MANAGEMENT (CONTINUED)

- CONFIGURATION IDENTIFICATION TECHNIQUES
- CONFIGURATION CONTROL MECHANISMS
- PROBLEM REPORTING SYSTEM
- AUDIT TRAIL TECHNIQUES AND VISIBILITY CONSIDERATIONS
  - APPROACH TO MONITOR THE ABOVE GROUND RULES
- RESPONSIBILITIES OF THE PRODUCT ASSURANCE MANAGER (PAM)
  - A) REVIEW AND APPROVE THE CONFIGURATION MANAGEMENT PLAN
  - B) SUPPORT THE PROJECT PERSONNEL, AS REQUESTED
  - C) PROVIDE GUIDANCE, AS REQUESTED
  - D) PHYSICAL MEDIA CONTROL, AS REQUESTED IN THE PLAN
  - E) PERIODIC AUDITS OF THE CONFIGURATION MANAGEMENT PROGRAM

## 18) SOFTWARE QUALITY ASSURANCE

- THE INDIVIDUAL RESPONSIBLE FOR QA REPORTS TO THE PROJECT MANAGER
- ASSIGNMENT TO THE JOB REQUIRES JOINT PROJECT MANAGER AND DIVISION PRODUCT ASSURANCE MANAGER APPROVAL
- FORMALITY AND CONTROL ARE TEMPERED AS IN CONFIGURATION MANAGEMENT
- ACTIVITIES ARE TO ASSURE INDEPENDENT ASSURANCE THAT THE STANDARDS OF QUALITY ARE MET VIA ONGOING REVIEW AND PERIODIC AUDITS
- A PLAN IS REQUIRED - A DEFAULT FORMAT IS AVAILABLE IN THE STANDARDS DOCUMENT





This section (A-1.6) is proprietary  
to the System Development Corp.  
and requires their permission  
before distribution

SYSTEM DEVELOPMENT CORPORATION STANDARDS AND GUIDELINES

SOFTWARE DEVELOPMENT MANUAL, JANUARY, 1976

1. INTRODUCTION
2. PLANNING PHASE  
(EACH PHASE: OVERVIEW, ACTIVITIES, REVIEWS)
3. REQUIREMENTS/PERFORMANCE PHASE
4. DESIGN PHASE
5. DEVELOPMENT PHASE
6. TEST AND ACCEPTANCE PHASE
7. OPERATIONS AND MAINTENANCE PHASE
8. PHASE INDEPENDENT ACTIVITIES  
(PROJECT, CONTRACT AND CONFIGURATION MANAGEMENT, QUALITY ASSURANCE)
9. GLOSSARY
10. BIBLIOGRAPHY

A-1.6 Characterization of SDC Software Management

(PROPRIETARY)

## SDC STANDARDS AND GUIDELINES

---

### AREA: TOP-LEVEL POLICY

REF: SECTIONS 1.7, 2.1, 3.1, 4.1, 5.1, 6.1, 7.1, 8.1

- THE SDM INCLUDES 6 PAGES OF BASIC STANDARDS (MANDATORY FOR ALL PROJECTS EXCEPT BY MANAGEMENT WAIVER)

### ● EXAMPLES OF MANDATORY PROVISIONS:

-- SDC STANDARD LIFECYCLE PHASES

-- BASIC DOCUMENTS:

-PROJECT PLAN

-CP PERFORMANCE SPEC

-CP DESIGN SPEC

-- PROGRAM PRODUCTION LIBRARY

-- TOP-DOWN/STRUCTURED DESIGN

-- PROGRAMMER'S REFERENCE MANUAL

-- USE OF HIGHER ORDER LANGUAGES

-- PROJECT NOTEBOOKS FOR EACH CPC

-- INDEPENDENT TEST AND INTEGRATION TEAM

-CPCI\* TEST PLAN

-CP DETAILED DEVELOPMENT SPEC (EACH CPC<sup>+</sup>)

-VERSION DESCRIPTION DOCUMENT

---

\* COMPUTER PROGRAM CONFIGURATION ITEM

+ COMPUTER PROGRAM COMPONENT

(PROPRIETARY)

SDC STANDARDS AND GUIDELINES

---

AREA: DEVELOPMENT PLAN

REF: SECTIONS 2.2.1-2.2.4, 8.2-8.4

- A PROJECT PLAN IS REQUIRED FOR ALL PROJECTS.. IT COMPRISES
  - SOFTWARE DEVELOPMENT PLAN
  - PROJECT WORK PLAN
  - PROJECT ORGANIZATION AND STAFFING PLAN
  - PROJECT TIME-PHASED COST BUDGET
  - DOCUMENTATION PLAN
  - CONFIGURATION MANAGEMENT PLAN
  - QUALITY ASSURANCE PLAN
  - PROJECT MONITORING AND CONTROL PROCEDURES
- THE SOFTWARE DEVELOPMENT PLAN IS THE BASIC, OVERVIEW PLANNING DOCUMENT
- THESE DOCUMENTS ARE PRODUCED DURING THE PLANNING PHASE AND ARE REVIEWED AT A FORMAL PROJECT PLAN REVIEW (PPR)
- AN UPDATED PROJECT PLAN IS REVIEWED AT ALL MAJOR REVIEW MILESTONES

(PROPRIETARY)

## SDC STANDARDS AND GUIDELINES

AREA: CONFIGURATION MANAGEMENT

REF: SECTIONS 2.2.6, 8.5

- AS WITH DOD, MAJOR FUNCTIONS INCLUDE CONFIGURATION IDENTIFICATION, CONTROL, AND STATUS ACCOUNTING AND REPORTING.
- A CONFIGURATION MANAGEMENT PLAN IS PREPARED FOR EACH SOFTWARE PROJECT.  
THE ITEMS UNDER CONTROL INCLUDE:
  - SYSTEM DESCRIPTION/SPEC
  - CP PERFORMANCE SPEC
  - CPCI(s)
  - CPCI TEST PLANS AND PROCEDURES
  - CP DESIGN SPEC
  - CP DETAILED DEVELOPMENT SPEC
  - CPC(s)
  - CP PRODUCT SPEC
  - VERSION DESCRIPTION DOCUMENTS

- THE BASELINE MANAGEMENT APPROACH IS USED AS WITH DOD:

SYSTEM DESCRIPTION/SPEC	→	FUNCTIONAL BASELINE
CP PERFORMANCE SPEC	→	ALLOCATED BASELINE
CP PRODUCT SPEC	→	PRODUCT BASELINE

- A PROGRAM PRODUCTION LIBRARY IS THE REPOSITORY FOR ALL SYSTEM DOCUMENTATION AND SOFTWARE PRODUCTS

(PROPRIETARY)

## SDC STANDARDS AND GUIDELINES

---

AREA: QUALITY ASSURANCE

REF: SECTIONS 2.2.7, 8.6

- A QUALITY ASSURANCE PLAN IS PREPARED AS PART OF THE OVERALL PROJECT PLAN
- TOPICS INCLUDE:
  - QUALITY STANDARDS
  - REVIEWS AND AUDITS
  - QA PERSONNEL RESPONSIBILITIES
  - INSPECTION AND TESTING
  - STORAGE MEDIA CONTROL PLAN
  - CUSTOMER ACCEPTANCE
  - QA POLICIES AND PROCEDURES
  - DISCREPANCY REPORTING
- QA IS ADMINISTERED AND MONITORED BY A QUALITY ASSURANCE BOARD (QAB) COMPRISED OF SDC PERSONNEL NOT DIRECTLY ASSOCIATED WITH THE PROJECT. (PEER REVIEW PROCESS).

(PROPRIETARY)

## SDC STANDARDS AND GUIDELINES

---

AREA: END-PRODUCT ACCEPTANCE

REF: SECTIONS 3.2.2.6, 6.2.3, 6.2.4, 6.3

- ACCEPTANCE CRITERIA ARE ESTABLISHED DURING THE REQUIREMENTS/PERFORMANCE PHASE AND ARE INCLUDED IN THE CP PERFORMANCE SPEC
- THE FORMAL QUALIFICATION TEST (FQT) IS THE PROCESS FOR DEMONSTRATING TO THE CUSTOMER CONFORMANCE WITH CONTRACTUAL REQUIREMENTS
- AN INTERNAL FORMAL QUALIFICATION REVIEW (FQR) IS CONDUCTED PRIOR TO FQT TO ASSURE READINESS
- FOLLOWING FQT, THE FUNCTIONAL AND PHYSICAL CONFIGURATION AUDITS ARE HELD (FCA, PCA); THE MEANINGS ARE AS FOR DoD.
- THE PHASING OF SYSTEM INTEGRATION, TEST AND ACCEPTANCE WITH SOFTWARE TEST AND ACCEPTANCE IS UNCLEAR

(PROPRIETARY)

SDC STANDARDS AND GUIDELINES

---

AREA: OPERATIONS AND MAINTENANCE PLAN

REF: SECTION 7

- ALTHOUGH THE OPERATIONS AND MAINTENANCE PHASE IS DISCUSSED,  
NO SEPARATE O&M PLAN IS IDENTIFIED.



SDC STANDARDS AND GUIDELINES

---

AREA: RESOURCE AND SCHEDULE ESTIMATION, MONITORING, AND CONTROL

REF: SECTIONS 2.2.2 - 2.2.4

- RESOURCE AND SCHEDULE ESTIMATES ARE PARTS OF THE PROJECT PLAN:
  - PROJECT WORK PLAN
    - PROJECT SCHEDULE, SOFTWARE PRODUCTION SCHEDULE
    - WORK BREAKDOWN STRUCTURE, TASK ASSIGNMENTS
    - COMPUTER TIME FORECAST
  - PROJECT ORGANIZATION AND STAFFING PLAN
  - PROJECT FINANCIAL PLAN AND BUDGET
- THESE PLANS ARE MAINTAINED THROUGHOUT THE PROJECT AND INCLUDE TASK COMPLETION AND COST DATA
- NO SPECIFIC GUIDELINES ARE INCLUDED FOR ESTIMATING SOFTWARE RESOURCE REQUIREMENTS

(PROPRIETARY)

SDC STANDARDS AND GUIDELINES

---

AREA: RISK ASSESSMENT

REF: NONE

- SDC APPROACH EMPHASIZES GOOD MANAGEMENT PRACTICES FOR PROJECT PLANNING, ORGANIZATION, AND CONTROL
- RISK ASSESSMENT IS NOT EXPLICITLY ADDRESSED

(PROPRIETARY)

## SDC STANDARDS AND GUIDELINES

---

AREA: DOCUMENTATION

REF: SECTION 2.2.5

- A SOFTWARE DOCUMENTATION PLAN IS REQUIRED AS PART OF THE PROJECT PLAN
- THE PLAN DEFINES AND PROVIDES STANDARDS FOR ALL PROJECT DOCUMENTATION BASED ON PROJECT REQUIREMENTS
- THE PLAN ALSO ESTABLISHES PROCEDURES FOR MAINTENANCE AND CONTROL OF SPECIFICATIONS AND OTHER CONTROLLED DOCUMENTS
- A MINIMUM SET OF DOCUMENTS IS PRESCRIBED:
  - CP PERFORMANCE SPEC
  - CP DESIGN SPEC
  - CP DETAILED DEVELOPMENT SPEC
  - CPCI TEST PLAN
  - CPCI TEST PROCEDURES
  - CPCI TEST RESULTS
  - OPERATING INSTRUCTIONS
  - PROGRAMMER'S REFERENCE MANUAL
  - VERSION DESCRIPTION DOCUMENT
  - (PROPRIETARY)

# SDC STANDARDS AND GUIDELINES

## AREA: DEVELOPMENT MILESTONES

REF: SECTIONS 2.3, 3.3, 4.3, 5.3, 6.3

- A PHASED SOFTWARE SYSTEM DEVELOPMENT LIFE CYCLE IS USED

PHASE	BASIC PRODUCTS	BASIC REVIEWS
PLANNING	PROJECT PLAN	PROJECT PLAN REVIEW (PPR)
REQUIREMENTS/PERFORMANCE	CP PERFORMANCE SPEC	REQUIREMENTS PERFORMANCE REVIEW (RPR)
DESIGN	CP DESIGN SPEC	SYSTEM DESIGN REVIEW (SDR)
	CPCI*/CPC <sup>†</sup> TEST PLANS	
DEVELOPMENT	CP DETAILED DEVELOPMENT SPECS	CPC DETAILED DESIGN REVIEWS
	PROGRAM SOURCE CODE	
	CPC TEST REPORTS	
TEST AND ACCEPTANCE	CP PRODUCT SPEC	FORMAL QUALIFICATION REVIEW (FQR)
	OPERATIONAL CPCI'S	FUNCTIONAL CONFIGURATION AUDIT (FCA)
	CPCI TEST REPORTS	PHYSICAL CONFIGURATION AUDIT (PCA)
	USER'S MANUALS	

(PROPRIETARY)

\*CPCI - COMPUTER PROGRAM CONFIGURATION ITEM

<sup>†</sup>CPC - COMPUTER PROGRAM COMPONENT

## SDC STANDARDS AND GUIDELINES

AREA: REQUIREMENTS SPECIFICATIONS

REF: SECTION 3

- THE COMPUTER PROGRAM PERFORMANCE SPECIFICATION IS THE BASIC REQUIREMENTS DOCUMENT (SIMILAR FORMAT TO DoD 85 (ALIAS PART 1) SPEC)
- IT IS GENERATED FROM A SYSTEM DESCRIPTION SPECIFICATION (OR OTHER INPUTS) VIA A PROCESS OF TOP-DOWN REFINEMENT.
- REQUIREMENTS/PERFORMANCE PHASE TASKS INCLUDE:
  - REVIEW SYSTEM SPEC AND CONSTRAINTS
  - ALLOCATE REQUIREMENTS TO FUNCTIONS AND MAN/MACHINE OPERATIONS
  - IDENTIFY AND DETAIL CP PERFORMANCE REQUIREMENTS
  - DEFINE MAN/MACHINE INTERFACES AND TASKS
  - IDENTIFY CP FUNCTIONAL INTERFACES
  - ESTABLISH ACCEPTANCE CRITERIA
  - PRODUCE SPEC
- BOTH INFORMAL REVIEW MEETINGS AND A FORMAL REQUIREMENTS PERFORMANCE REVIEW (RPR) ARE CONDUCTED

(PROPRIETARY)

## SDC STANDARDS AND GUIDELINES

### AREA: DESIGN SPECIFICATIONS

REF: SECTIONS 4, 5.2.1

- THE COMPUTER PROGRAM DESIGN SPECIFICATION IS THE BASIC PRELIMINARY DESIGN DOCUMENT FOR EACH CPCI (SIMILAR TO DoD C5 (ALIAS PART 2) SPEC, BUT WITH LESS DETAIL ON INDIVIDUAL CPCs\*, AND MORE DETAIL ON SYSTEM RESOURCE UTILIZATION, THE DATA BASE, AND CPC TEST PLANS)
- DESIGN PHASE TASKS INCLUDE:
  - DEVELOP TOP DOWN STRUCTURE AND IDENTIFY CPCs
  - ALLOCATE REQUIREMENTS/PERFORMANCE
  - ESTABLISH TRACEABILITY
  - IDENTIFY STORAGE, DATA BASE, AND TIMING REQUIREMENTS
  - PROVIDE CENTRALIZED DATA DEFINITION AND CONTROL
  - DEVELOP IMPLEMENTATION APPROACH
  - DEVELOP CPC TEST PLAN
  - IDENTIFY SUPPORT AND TEST TOOLS
  - PRODUCE DOCUMENTATION (CP DESIGN SPEC AND PROGRAMMERS REFERENCE MANUAL)
- A FORMAL SYSTEM DESIGN REVIEW (SDR) IS CONDUCTED (SIMILAR TO DoD PDR) ON THE CP DESIGN SPEC
- THE COMPUTER PROGRAM DETAILED DEVELOPMENT SPECIFICATION IS THE BASIC DETAILED DESIGN DOCUMENT FOR EACH CPC (SIMILAR TO DoD C5, SECTION 3), AND IS REVIEWED AT A CPC DETAILED DESIGN REVIEW (SIMILAR TO DoD CDR)

\*NOTE A CPC CAN BE A FAIRLY LARGE UNIT OF CODE (SEVERAL THOUSAND LINES)

## SDC STANDARDS AND GUIDELINES

AREA: PROGRAM DEVELOPMENT

REF: SECTION 5

- CODING OF THE CPCs PROCEEDS IN ACCORDANCE WITH THE PROGRAMMER'S REFERENCE MANUAL  
(PRODUCED IN THE DESIGN PHASE)
- THE PROGRAMMER'S REFERENCE MANUAL INCLUDES:
  - CODING STANDARDS AND CONVENTIONS
  - OS INTERFACE PROCEDURES
  - JCL CONVENTIONS
  - PROGRAM PRODUCTION LIBRARY PROCEDURES
  - ERROR REPORTING AND CONTROL PROCEDURES
  - SUPPORT TOOL AVAILABILITY AND USE
  - INTEGRATION STRATEGY
- PROJECT-SPECIFIC CODING STANDARDS ADDRESS:
  - PROGRAM ORGANIZATION
  - PROGRAM STRUCTURES
  - IDENTATION
  - NAMING CONVENTIONS
  - COMMENTS
- CODING IS FOLLOWED BY DESK CHECKING AND CPC TESTING
- FOLLOWING SUCCESSFUL CPC TESTING, THE CPC IS TRANSITIONED INTO CPC1 TESTING

(PROPRIETARY)

SDC STANDARDS AND GUIDELINES

---

AREA: TEST, EVALUATION, AND DEMONSTRATION

REF: SECTIONS 5.2.4, 6

- THREE LEVELS OF TESTING ARE IDENTIFIED
  - CPC TESTING (DEVELOPMENT PHASE; DEVELOPMENT TEAM)
  - CPCI TESTING
  - ACCEPTANCE/INSTALLATION TESTING } (TEST AND ACCEPTANCE PHASE; SYSTEM TEST TEAM)
- CPC TESTING ACTIVITIES INCLUDE:
  - DEVELOPING THE CPC TEST PLANS AND PROCEDURES
  - CONDUCTING THE TESTS, CORRECTING ERRORS, AND REPORTING RESULTS
- CPCI TESTING ACTIVITIES INCLUDE:
  - DEVELOPING THE CPCI TEST PLAN (INCLUDING FORMAL QUALIFICATION TEST (FQT)) AND PROCEDURES
  - CONDUCTING THE TESTS, HANDLING DISCREPANCIES, AND REPORTING RESULTS
  - CONDUCTING THE FORMAL QUALIFICATION TEST
- SOFTWARE/HARDWARE INTEGRATION TESTING IS NOT ADDRESSED IN MUCH DETAIL

(PROPRIETARY)



## SDC STANDARDS AND GUIDELINES

---

AREA: END-PRODUCT USE

REF: SECTION 7.2.1

- USER DOCUMENTATION IS PRODUCED DURING THE DEVELOPMENT PHASE AND REVIEWED DURING THE TEST AND ACCEPTANCE PHASE. FINAL REVIEW OCCURS AT THE PHYSICAL CONFIGURATION AUDIT.
- OPERATION SUPPORT TASKS INCLUDE
  - DIAGNOSIS OF SOFTWARE PROBLEMS
  - INSTALLATION OF SUCCESSIVE RELEASES
  - CONFIGURATION CONTROL SUPPORT
  - OPERATOR AND USER TRAINING

(PROPRIETARY)

SDC STANDARDS AND GUIDELINES

---

AREA: MAINTENANCE

REF: SECTION 7

- THE O&M PHASE IS INITIATED FOLLOWING SYSTEM ACCEPTANCE AND INCLUDES AS INPUTS:
  - OPERATIONS AND SUPPORT COMPUTER PROGRAMS
  - CP PERFORMANCE AND PRODUCT SPECS
  - TEST PLANS/PROCEDURES/REPORTS
  - USER'S MANUALS, PROJECT NOTEBOOKS, PROJECT PLAN
  - VERSION DESCRIPTION DOCUMENT
- MAINTENANCE TASKS INCLUDE:
  - ERROR CORRECTION
  - TESTS ON NEW RELEASES
  - CONFIGURATION CONTROL RESPONSIBILITY
- PHASE OUTPUTS INCLUDE:
  - PROBLEM REPORTS
  - UPDATED CP TAPES
  - UPDATED DOCUMENTATION
- THE PROGRAM PRODUCTION LIBRARY IS NOT DISCUSSED

(PROPRIETARY)

## SDC STANDARDS AND GUIDELINES

---

AREA: DATA COLLECTION

REF: SECTIONS 2, 8.2.4

- THE COLLECTION OF SCHEDULE, COMPUTER TIME UTILIZATION, AND COST DATA IS PRESCRIBED
- SCHEDULE DATA (ACTUAL VS ESTIMATED TASK COMPLETIONS) IS REPORTED IN GRAPHICAL FORM (45-DEGREE CHARTS)
- NO SPECIFIC INSTRUCTIONS ARE GIVEN REGARDING EITHER PRODUCTIVITY OR ERROR DATA ELEMENT DEFINITIONS

(PROPRIETARY)

SDC STANDARDS AND GUIDELINES

AREA: PROCUREMENT; CERTIFICATION

REF: NONE

- PROCUREMENT AND SUBCONTRACTING IS NOT DISCUSSED
- CERTIFICATION BEYOND NORMAL VERIFICATION AND VALIDATION IS NOT DISCUSSED.



DOCUMENTS REVIEWED

● AIR FORCE SOFTWARE ACQUISITION MANAGEMENT GUIDEBOOK SERIES

- SAMG 1 OVERVIEW OF THE SERIES (AD A055575)
- SAMG 2 REGULATIONS, SPECIFICATION & STANDARDS (AD A016401)
- SAMG 3 CONTRACTING FOR SOFTWARE ACQUISITION (AD A020444)
- SAMG 4 MONITORING AND REPORTING SOFTWARE DEVELOPMENT STATUS (AD A016488)
- SAMG 5 STATEMENT OF WORK PREPARATION (AD A035924)
- SAMG 6 REVIEWS AND AUDITS (AD A052567)
- SAMG 7 CONFIGURATION MANAGEMENT (AD A047308)
- SAMG 8 COMPUTER PROGRAM DEVELOPMENT SPECIFICATION (AD A055573)
- SAMG 9 SOFTWARE DOCUMENTATION REQUIREMENTS (AD A027051)
- SAMG 10 VERIFICATION (AD A048577)
- SAMG 11 VALIDATION AND CERTIFICATION (AD A053039)
- SAMG 12 SOFTWARE MAINTENANCE (AD A053040)
- SAMG 13 SOFTWARE QUALITY ASSURANCE (AD A047318)
- SAMG 14 SOFTWARE COST ESTIMATING AND MEASURING (AD A055574)
- SAMG 15 SOFTWARE DEVELOPMENT AND MAINTENANCE FACILITIES (AD A038234)
- SAMG 16 LIFE CYCLE EVENTS (AD A037115)

APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

DOCUMENTS REVIEWED (CONT'D)

- DOD DIRECTIVES AND INSTRUCTIONS

DODD 5000.29, "MANAGEMENT OF COMPUTER RESOURCES IN MAJOR DEFENSE SYSTEMS"

DODI 5000.31, "INTERIM LIST OF DOD APPROVED HIGH ORDER PROGRAMMING LANGUAGES"

- MIL-STANDARDS

MIL-STD-483, "CONFIGURATION MANAGEMENT PRACTICES"

MIL-STD-490, "SPECIFICATION PRACTICES"

MIL-STD-1521A, "TECHNICAL REVIEWS AND AUDITS"

MIL-S-52779, "SOFTWARE QUALITY ASSURANCE PROGRAM REQUIREMENTS"

- AIR FORCE REGULATIONS

AFR 800-14, VOL. 2, "ACQUISITION AND SUPPORT PROCEDURES FOR COMPUTER RESOURCES IN SYSTEMS"



## APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

### AREA: TOP-LEVEL POLICY

REF: DODD 5000.29, "MANAGEMENT OF COMPUTER RESOURCES IN MAJOR DEFENSE SYSTEMS"  
DODI 5000.31, "INTERIM LIST OF DOD APPROVED HIGH ORDER PROGRAMMING LANGUAGES"  
AFR-800-14, VOL. 2, "ACQUISITION AND SUPPORT PROCEDURES FOR COMPUTER RESOURCES IN SYSTEMS"

- DoD AND AIR FORCE TOP-LEVEL POLICY STATEMENTS ARE BROADER AND LESS SPECIFIC THAN NMI 2410.6
- NMI 2410.6 FOCUSES ON THE SOFTWARE MANAGEMENT PLAN CONTENTS; DOD/AF POLICY ADDRESSES THE MANAGEMENT PROCEDURES DIRECTLY
- IN ADDITION TO A COMPUTER PROGRAM DEVELOPMENT PLAN (CPDP) (SIMILAR TO THE NASA SMP), DOD/AF POLICY STRESSES THE DEVELOPMENT OF A COMPUTER RESOURCES INTEGRATED SUPPORT PLAN (CRISP) FOR TOTAL LIFE CYCLE PLANNING
- THE CPDP IS TYPICALLY A CONTRACTOR RESPONSIBILITY, WHEREAS THE CRISP IS A PROGRAM OFFICE RESPONSIBILITY
- TOPICS ADDRESSED BY DOD/AF TOP LEVEL POLICY (BUT NOT NMI 2140.6):
  - SOFTWARE CONSIDERATION EARLY IN THE LIFE CYCLE
  - SOFTWARE COMPONENTS AS DELIVERABLE CONFIGURATION ITEMS
  - SUPPORT SOFTWARE AS DELIVERABLE
  - USE OF DOD-APPROVED HIGH ORDER LANGUAGES



AREA: DEVELOPMENT PLAN

REF: SAMG 9, "SOFTWARE DOCUMENTATION REQUIREMENTS"  
DI-S-30567A, "COMPUTER PROGRAM DEVELOPMENT PLAN"

- THE COMPUTER PROGRAM DEVELOPMENT PLAN (CPDP) IS THE BASIC MANAGEMENT PLANNING DOCUMENT FOR SOFTWARE DEVELOPMENT.
- IT IS NORMALLY CONTRACTOR GENERATED EITHER IN RESPONSE TO AN RFP OR AS A CONTRACT DELIVERABLE DOCUMENT.
- CERTAIN ASPECTS OF THE DEVELOPMENT PLAN MAY BE DEFINED IN MORE DETAIL IN OTHER DOCUMENTS: CONFIGURATION MANAGEMENT PLAN, TEST AND EVALUATION MASTER PLAN, QUALITY ASSURANCE PLAN.
- THIS DOCUMENT CORRESPONDS MOST CLOSELY TO THE SOFTWARE MANAGEMENT PLAN REFERRED TO IN NM12410.6.





**COMPUTER PROGRAM DEVELOPMENT PLAN OUTLINE (DI-S-30567A)**

---

- 1. REQUIREMENTS ASSESSMENT SUMMARY**
- 2. PROJECT OBJECTIVES**
- 3. WORK DEFINITION**
- 4. WORK SCHEDULE**
- 5. ACTIVITY NETWORK**
- 6. ORGANIZATION**
- 7. RESOURCE ALLOCATION**
- 8. ENGINEERING STANDARDS**
- 9. DESIGN ASSURANCE TECHNIQUES**
- 10. DETAILED DESIGN, CODING, AND CHECKOUT**
- 11. DEVELOPMENT TEST AND EVALUATION**
- 12. SYSTEM TEST AND EVALUATION**
- 13. ANOMALY CONTROL**
- 14. MANAGEMENT CONTROLS**
- 15. DOCUMENTATION**
- 16. CONFIGURATION MANAGEMENT**
- 17. VENDOR/GFE COMPUTER RESOURCES**
- 18. SUPPORT RESOURCES FOR THE DEPLOYMENT PHASE**



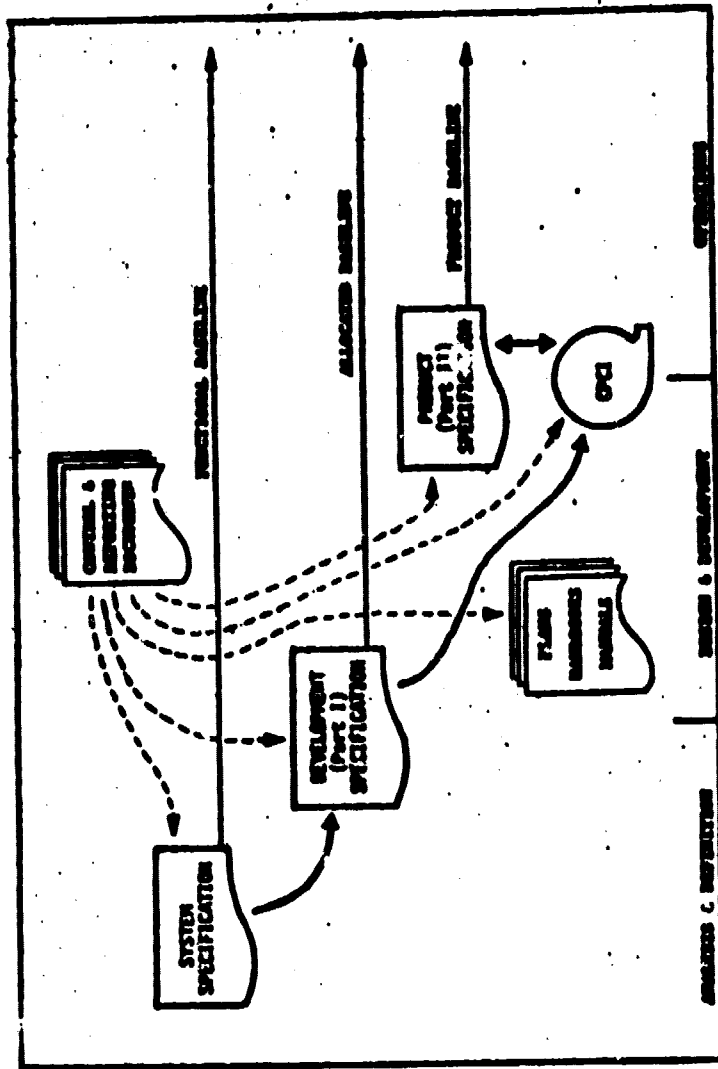
## APPLICABILITY OF DOD STANDARDS AND GUIDELINES

### AREA: CONFIGURATION MANAGEMENT

REF: SAMG 7, "COMPUTER PROGRAM CONFIGURATION MANAGEMENT"  
MIL-STD-483, "CONFIGURATION MANAGEMENT PRACTICES"

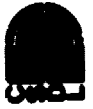
- CONFIGURATION MANAGEMENT WITHIN THE DOD HAS THREE MAIN FUNCTIONS:
  - CONFIGURATION IDENTIFICATION
  - CONFIGURATION CONTROL
  - STATUS ACCOUNTING AND REPORTING
- BASELINE MANAGEMENT IS THE BASIC APPROACH TO CONFIGURATION IDENTIFICATION AND CONTROL, WITH AUTHENTICATED SPECIFICATION DOCUMENTS PROVIDING THE CONFIGURATION BASELINE. A COMPUTER PROGRAM IS CONTROLLED BY CONTROLLING THE DOCUMENTATION WHICH DESCRIBES IT.
- CHANGE CONTROL IS ACCOMPLISHED BY REQUIRING PROCURING ACTIVITY APPROVAL OF ENGINEERING CHANGE PROPOSALS (ECP's) WITH ACCOMPANYING SPECIFICATION CHANGE NOTICES (SCN's).
- STATUS ACCOUNTING AND REPORTING IS ACCOMPLISHED USING SPECIALIZED DOCUMENTS: CONFIGURATION INDEX, CHANGE STATUS REPORT, AND VERSION DESCRIPTION DOCUMENT.
- CONTRACTOR CONTROL OF BASELINED MATERIALS IS ACCOMPLISHED USING AUTOMATED PROGRAM SUPPORT LIBRARY (PSL) TOOLS.
- A CONTRACTOR-PREPARED CONFIGURATION MANAGEMENT PLAN IS REQUIRED TO DOCUMENT THESE PROCEDURES.

## CONFIGURATION IDENTIFICATION: BASELINE MANAGEMENT

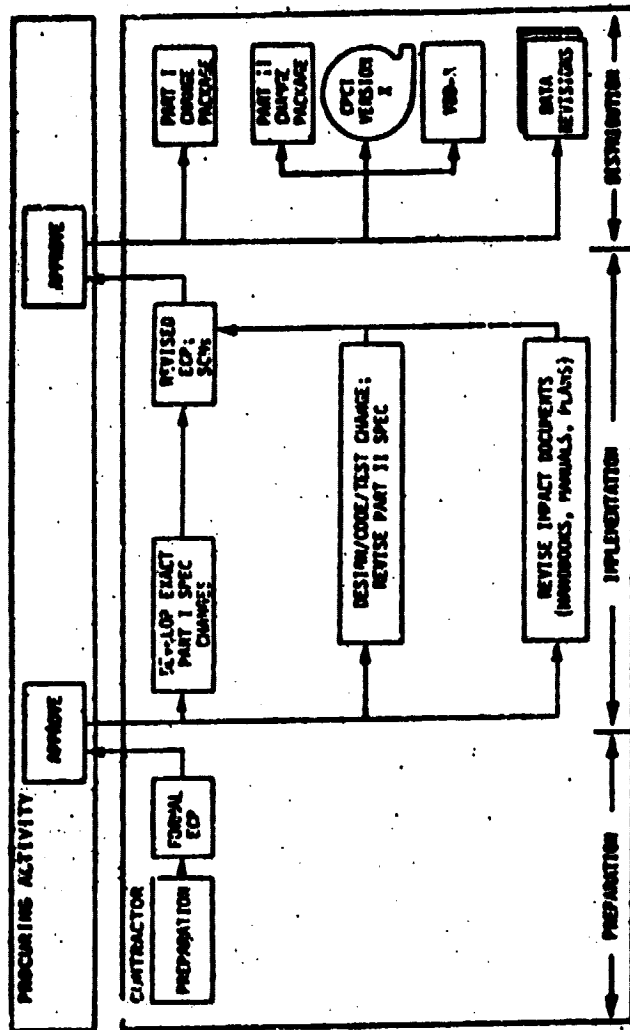


SOURCE: AF ESD-TR-77-254

- THE CONTROLLED ITEM IS DESIGNATED A COMPUTER PROGRAM CONFIGURATION ITEM (CPCI)
- EACH CPCI HAS AN ALLOCATED BASELINE AND A PRODUCT BASELINE ASSOCIATED WITH IT (PRODUCT SPEC INCLUDES CODE LISTING)
- DURING DEVELOPMENT, PRIMARY PROGRAM OFFICE CONCERN IS WITH CONTROLLING THE DOCUMENTATION



# CONFIGURATION CONTROL: CHANGE PROCESSING (TWO STEPS)



SOURCE: AF ESD-77-254

- CHANGE PROCESSING FORMS:
  - ENGINEERING CHANGE PROPOSAL (ECP)
  - SPECIFICATION CHANGE NOTICE (SCN)
- CLASS I CHANGES (AFFECT PART I SPEC, SCHEDULE, OR COST) REQUIRE APPROVAL BEFORE IMPLEMENTATION
- CLASS II CHANGES (MINOR CHANGES) MUST BE REPORTED FOR CONCURRENCE WITH CLASSIFICATION

ORIGINAL PAGE IS  
OF POOR QUALITY

## STATUS ACCOUNTING AND REPORTING

---

- STATUS ACCOUNTING AND REPORTING IS ACCOMPLISHED USING:
  - CONFIGURATION INDEX (MONTHLY)
  - CHANGE STATUS REPORT (MONTHLY)
  - VERSION DESCRIPTION DOCUMENT (ONCE PER CPCI RELEASE)
- CONFIGURATION INDEX - LISTS THE CURRENT BASIC ISSUE AND SCN'S (TITLE, DATE) FOR ALL CPCI DOCUMENTATION SUBJECT TO CONFIGURATION MANAGEMENT
- CHANGE STATUS REPORT - LISTS THE CURRENT STATUS FOR ALL ECP'S WHICH ARE ACTIVE AT A GIVEN TIME (ECP'S ARE DROPPED FROM THE LIST AFTER FINAL DISPOSITION)
- VERSION DESCRIPTION DOCUMENT - ACCOMPANIES THE DELIVERED CPCI AND ASSOCIATED DOCUMENTATION TO IDENTIFY THE ITEMS DELIVERED AND PROVIDE FIELD PERSONNEL WITH ANY NEEDED ADDITIONAL DATA.



CONFIGURATION MANAGEMENT PLAN OUTLINE (MIL-STD 483)

---

1. ORGANIZATION
2. CONFIGURATION IDENTIFICATION
3. CONFIGURATION CONTROL
4. CONFIGURATION STATUS ACCOUNTING
5. SUBCONTRACTOR/VENDOR CONTROL
6. PROGRAM PHASING
7. MANAGEMENT INTEGRATION OF CONFIGURATION MANAGEMENT
8. CONFIGURATION AUDITS



APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: QUALITY ASSURANCE

REF:

SAMG 4, "MONITORING AND REPORTING SOFTWARE DEVELOPMENT STATUS"

SAMG 13, "SOFTWARE QUALITY ASSURANCE"

MIL-S-52779, "SOFTWARE QUALITY ASSURANCE PROGRAM REQUIREMENTS"

- THE DOD STANDARD MIL-S-52779 SPECIFIES REQUIREMENTS FOR A CONTRACTOR SOFTWARE QUALITY ASSURANCE (SQA) PROGRAM
- THE QUALITY ASSURANCE PROGRAM PLAN CAN BE DOCUMENTED SEPARATELY (NO OUTLINE IS PROVIDED) OR AS PART OF THE COMPUTER PROGRAM DEVELOPMENT PLAN
- TOPICS TO BE ADDRESSED INCLUDE:
  - PROGRESS MONITORING
  - CONFIGURATION MANAGEMENT
  - TESTING
  - DISCREPANCY REPORTING
  - CORRECTIVE ACTION
  - SUBCONTRACTOR CONTROL
  - LIBRARY CONTROLS
  - COMPUTER PROGRAM DESIGN
  - DOCUMENTATION
  - REVIEWS AND AUDITS
  - TOOLS, TECHNIQUES, METHODOLOGIES

- THE INFORMATION IN THE CURRENT SQA GUIDEBOOK AT PRESENT DOES NOT ADDRESS QUANTITATIVE APPROACHES TO SOFTWARE QUALITY SPECIFICATION AND MEASUREMENT

SOFTWARE QUALITY ASSURANCE

STATUS FACTORS

QUALITY OF DOCUMENTATION

STABILITY OF REQUIREMENTS BASELINE

INTERFACES

PROGRAMMING LANGUAGES

PROGRAMMING PRACTICES AND STANDARDS

PROJECT COMPLEXITY

OPERATING SYSTEMS

DATA MANAGEMENT

PERSONNEL

QUALITY OF DEVELOPMENT FACILITY

PROJECT MANAGEMENT

NONSUBJECTIVE DATA: PERSONNEL, CODING AND DOCUMENTATION, COMPUTER USAGE, TRENDS





APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: END-PRODUCT ACCEPTANCE

REF: SAMG 6, "REVIEWS AND AUDITS"  
SAMG 11, "VALIDATION AND CERTIFICATION"

- THE FUNCTIONAL AND PHYSICAL CONFIGURATION AUDITS (FCA, PCA) ARE THE BASIC END-PRODUCT ACCEPTANCE PROCEDURES FOR DOD ACQUISITIONS (AT THE COMPONENT LEVEL)
- THE FCA VERIFIES THAT THE COMPLETED CPCI\* SATISFIES THE REQUIREMENTS OF THE CPCI DEVELOPMENT SPEC
- THE PCA VERIFIES THAT THE CPCI PRODUCT SPEC (INCLUDING PROGRAM LISTINGS) AND OTHER SUPPORT DOCUMENTATIONS ACCURATELY AND ADEQUATELY DESCRIBES THE DELIVERED CPCI.
- THE SIGNING OF THE DD FORM 250 AT THE SUCCESSFUL CONCLUSION OF THE PCA INDICATES GOVERNMENT ACCEPTANCE OF THE DELIVERED CPCI.
- FOLLOWING FCA AND PCA, THE SYSTEM COMPONENTS MUST BE INTEGRATED AND TESTED (SYSTEM DEVELOPMENT TEST AND EVALUATION)

---

\*COMPUTER PROGRAM CONFIGURATION ITEM



APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: OPERATIONS AND MAINTENANCE PLAN

REF: SAMG 9, "SOFTWARE DOCUMENTATION REQUIREMENTS"

AFR 800-14, VOL. 2, "ACQUISITION AND SUPPORT PROCEDURES FOR COMPUTER RESOURCES IN SYSTEM"

- THE COMPUTER RESOURCES INTEGRATED SUPPORT PLAN (CRISP) IS THE BASIC PLANNING DOCUMENT TO IDENTIFY THE COMPUTER RESOURCES NECESSARY TO SUPPORT THE SYSTEM AFTER END-PRODUCT ACCEPTANCE. IT IS THE BASIC AGREEMENT BETWEEN THE SUPPORTING (DEVELOPMENT) AND USING ORGANIZATIONS.
- IT IS NORMALLY GENERATED BY THE PROGRAM OFFICE PRIOR TO THE SOFTWARE DEVELOPMENT RFP RELEASE.
- A KEY ELEMENT IS THE IDENTIFICATION IN THE CRISP OF SUPPORT SOFTWARE DELIVERABLES NEEDED FOR SOFTWARE MAINTENANCE.
- A PLAN LIKE THE CRISP IS USEFUL IN IDENTIFYING AND REDUCING THE COSTS ASSOCIATED WITH SYSTEM DEPLOYMENT, SUPPORT, AND MAINTENANCE (OFTEN OVERLOOKED AND NORMALLY VERY LARGE).



## APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: RESOURCE AND SCHEDULE ESTIMATION, MONITORING, AND CONTROL

REF: SAMG 14, "SOFTWARE COST ESTIMATING AND MEASURING"

- THIS GUIDEBOOKS PROVIDES A REVIEW OF COST ESTIMATION METHODS AND ISSUES  
(HAHN & STONE, PUTNAM, TECOLOTE)
- NO SPECIFIC METHODS ARE RECOMMENDED



AREA: RISK ASSESSMENT

REF: SAMG 3, "CONTRACTING FOR SOFTWARE ACQUISITION"

SAMG 14, "SOFTWARE COST ESTIMATING AND MEASURING"

- TYPES OF RISK ARE IDENTIFIED AS
  - COST RISKS
  - TECHNOLOGICAL RISKS (E.G., A PROBLEM BEYOND THE STATE-OF-THE-ART OF  
COMPUTER TECHNOLOGY )
- THE SAM GUIDEBOOKS DO NOT PROVIDE MUCH GUIDANCE IN MANAGING HIGH RISK  
DEVELOPMENTS



APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: DOCUMENTATION

REF: SAMG 9, "SOFTWARE DOCUMENTATION REQUIREMENTS"

- DOD STRESSES NEED TO MATCH DOCUMENTATION PLANS TO PROJECT MAGNITUDE, COMPLEXITY, DURATION, TECHNICAL RISK, AND SCHEDULE
- LIST OF MINIMUM SET OF DOCUMENTS
- CONTRACTUAL ASPECTS OF ACQUIRING DOCUMENTATION
  - CONTRACT DATA REQUIREMENTS LIST (CDRL) SPECIFIES DOCUMENTS REQUIRED (CONTENT, FORMAT, DELIVERY DATE, NUMBER OF COPIES)
  - CDRL REFERENCES A DATA ITEM DESCRIPTION (DID) TO SPECIFY DETAILED CONTENT AND FORMAT



## DOCUMENTATION STANDARDS

### A MINIMUM SET OF DOCUMENTS

#### MANAGEMENT

- \*COMPUTER RESOURCES INTEGRATED SUPPORT PLAN (CRISP)
- \*COMPUTER PROGRAM DEVELOPMENT PLAN (CPDP)

#### USER REQUIREMENTS

##### USERS MANUAL

#### SPECIFICATIONS

- SYSTEM SPECIFICATION (MIL-STD 490, TYPE A)
- DEVELOPMENT SPECIFICATION (MIL-STD 490, TYPE B5)
- PRODUCT SPECIFICATION (MIL-STD 490, TYPE C5)

#### TESTING

##### CPCI TEST PLAN/PROCEDURES

##### CPCI TEST REPORT

##### SYSTEM TEST PLAN/PROCEDURES

##### SYSTEM TEST REPORT

- \*REQUIRED UNDER AFSC POLICY

ORIGINAL PAGE 1  
OF FOUR



APPLICABILITY OF DOD STANDARDS AND GUIDELINES

AREA: DEVELOPMENT MILESTONES

REF: SAMG 4, "MONITORING AND REPORTING SOFTWARE DEVELOPMENT STATUS"

SAMG 6, "REVIEWS AND AUDITS"

SAMG 16, "LIFE CYCLE EVENTS"

- BASIC DOD LIFE CYCLE IS THE SYSTEM ACQUISITION LIFE CYCLE WITH A WELL-DEFINED SET OF REVIEWS AND AUDITS AS MILESTONES (A SEPARATE SOFTWARE LIFE CYCLE IS ALSO IDENTIFIED IN AFR 800-14, VOL. 2)

- DOD NOMENCLATURE IS WIDELY USED, PARTICULARLY FOR REVIEWS AND AUDITS:

MILESTONE

DOCUMENTS TO BE REVIEWED (SIMPLIFIED)

SYSTEM REQUIREMENTS REVIEW (SRR)

REQUIREMENTS ANALYSES, SYSTEM SPEC (TYPE A)

SYSTEM DESIGN REVIEW (SDR)

DRAFT DEVELOPMENT SPEC (TYPE B5 FOR COMPUTER PROGRAMS)

PRELIMINARY DESIGN REVIEW (PDR)

PRELIMINARY DESIGN DATA

CRITICAL DESIGN REVIEW (CDR)

DRAFT PRODUCT SPEC (TYPE C5 FOR COMPUTER PROGRAMS)

FUNCTIONAL CONFIGURATION AUDIT (FCA)

COMPUTER PROGRAM VS. DEVELOPMENT SPEC

PHYSICAL CONFIGURATION AUDIT (PCA)

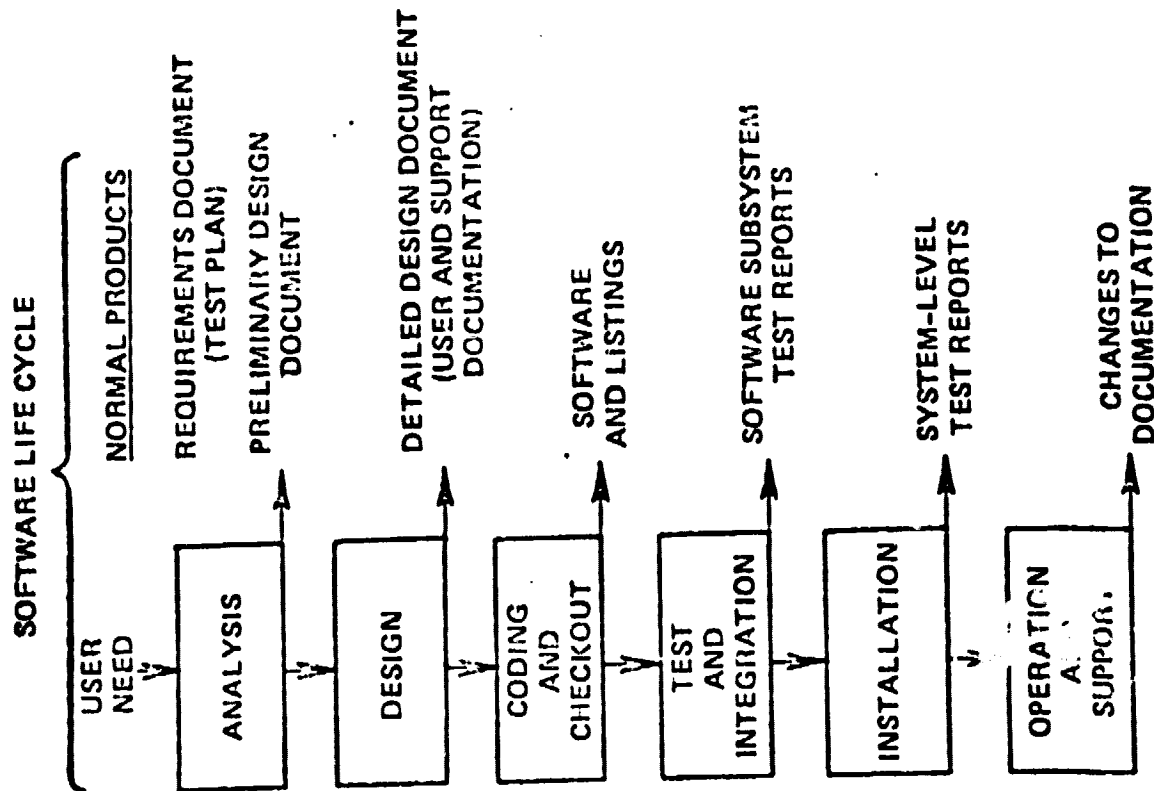
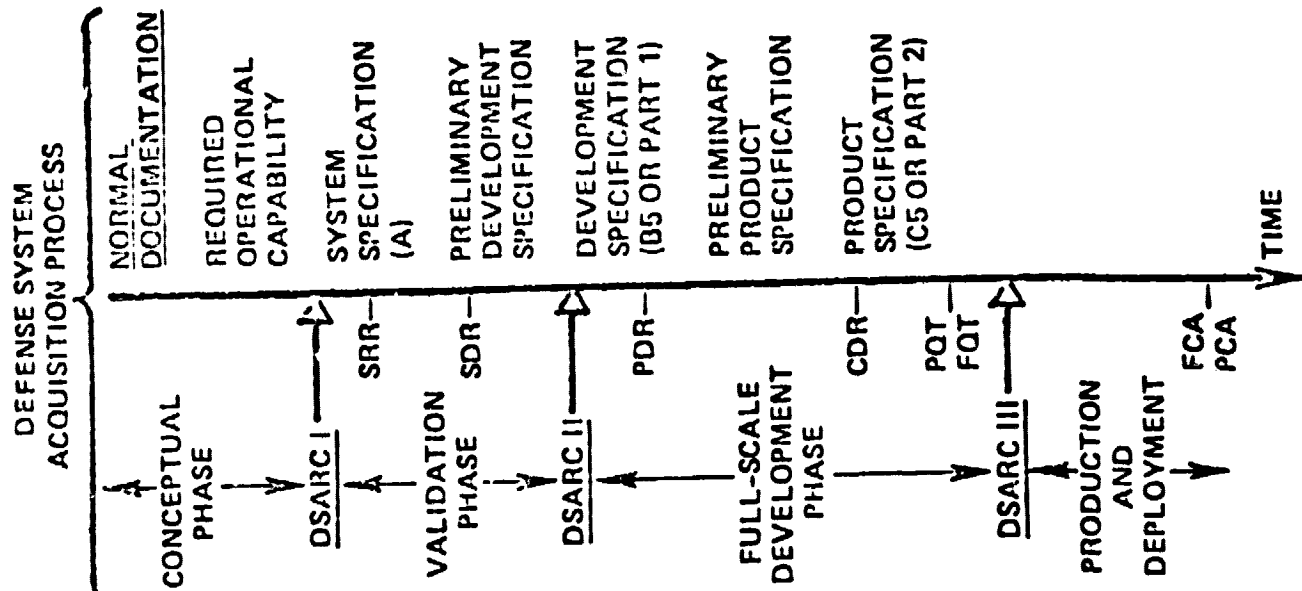
COMPUTER PROGRAM VS. PRODUCT SPEC

NOTE: DEVELOPMENT SPEC EMPHASIS IS REQUIREMENTS (DESIGN-TO-SPEC)

PRODUCT SPEC EMPHASIS IS DESIGN (CODE-TO; AS-CODED SPEC)

- THE ITEM PASSING THROUGH THE LIFE CYCLE IS DESIGNATED A COMPUTER PROGRAM CONFIGURATION ITEM (CPCI); EACH CPCI (POTENTIALLY) HAS ITS OWN LIFE CYCLE; THE EARLY IDENTIFICATION OF CPCI'S IS CRUCIAL

# LIFE-CYCLE PHASES





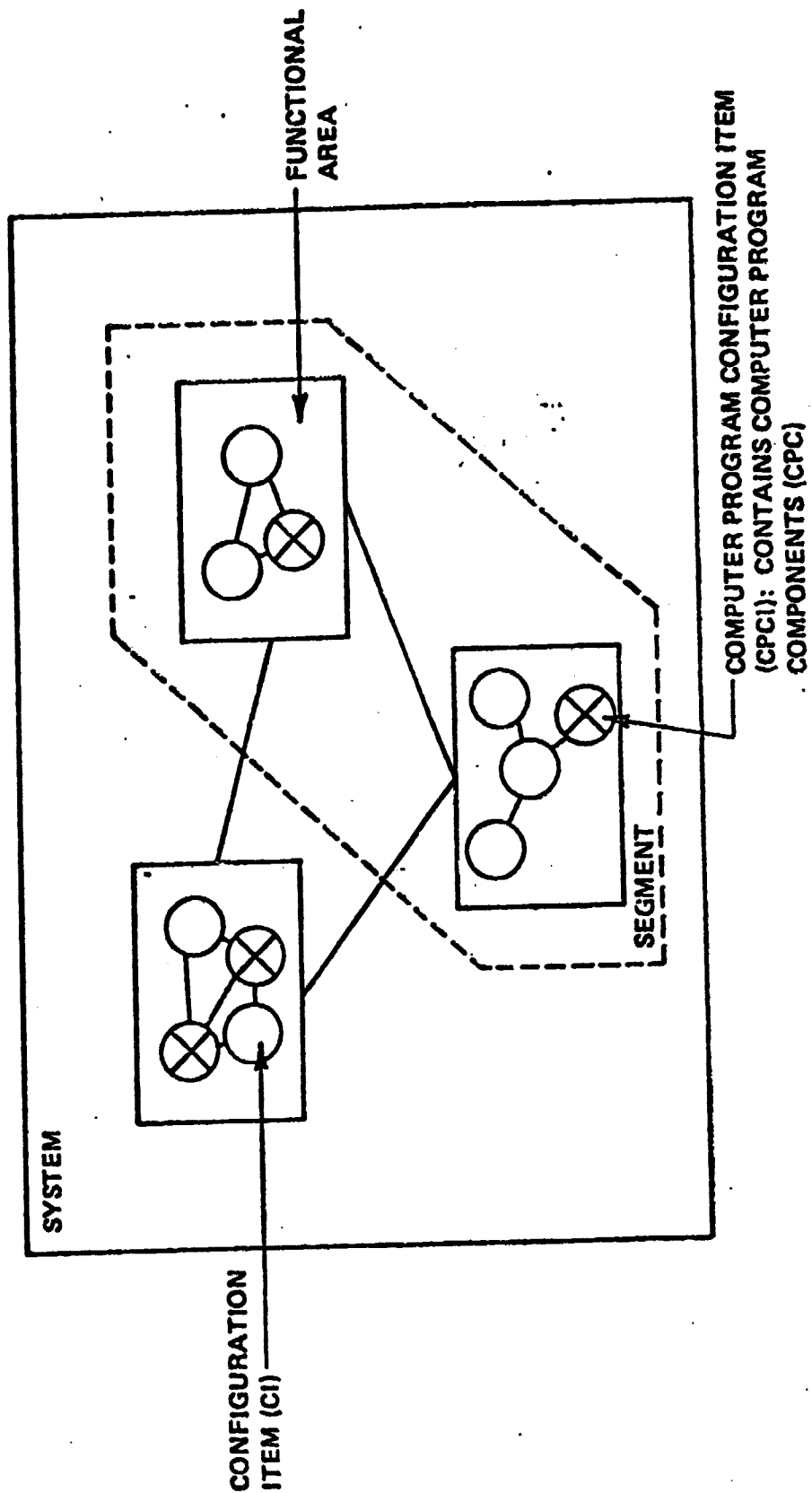
**LIFE-CYCLE PHASES (CONT.)**

**NOMENCLATURE**

DSARC — DEFENSE SYSTEM ACQUISITION REVIEW COUNCIL (DODD 5000.1)  
ROC — REQUIRED OPERATIONAL CAPABILITY  
SRR — SYSTEM REQUIREMENTS REVIEW  
SDR — SYSTEM DESIGN REVIEW  
PDR — PRELIMINARY DESIGN REVIEW  
CDR — CRITICAL DESIGN REVIEW  
PQT — PRELIMINARY QUALIFICATION TEST  
FQT — FORMAL QUALIFICATION TEST  
FCA — FUNCTIONAL CONFIGURATION AUDIT  
PCA — PHYSICAL CONFIGURATION AUDIT



# SYSTEM COMPONENTS



## APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: REQUIREMENTS SPECIFICATIONS

REF: SAMG 8, "COMPUTER PROGRAM DEVELOPMENT SPECIFICATION"

MIL-STD-483, "CONFIGURATION MANAGEMENT PRACTICES"

MIL-STD-490, "SPECIFICATION PRACTICES"

- THE DEVELOPMENT SPECIFICATION IS THE BASIC REQUIREMENTS DOCUMENT FOR SOFTWARE (ALSO KNOWN AS TYPE B5 SPEC, PART 1 SPEC, OR ALLOCATED BASELINE)
- DOD PRACTICE EMPHASIZES THE RELATION BETWEEN THE SYSTEM SPECIFICATION (ALSO KNOWN AS TYPE A SPEC, OR FUNCTIONAL BASELINE) AND THE DEVELOPMENT SPECIFICATIONS FOR SYSTEM ELEMENTS
- THE SYSTEM SPECIFICATION CONTAINS REQUIREMENTS FOR THE SYSTEM AS A WHOLE, AND THE ALLOCATION OF THOSE REQUIREMENTS TO SYSTEM ELEMENTS (CALLED CONFIGURATION ITEMS FOR HARDWARE OR COMPUTER PROGRAM CONFIGURATION ITEMS FOR SOFTWARE)
- OUTLINE FOR SOFTWARE DEVELOPMENT SPECIFICATION IS WIDELY USED AND WOULD BE APPROPRIATE FOR NASA SYSTEMS



APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: DESIGN SPECIFICATIONS

REF: SAMG 9, "SOFTWARE DOCUMENTATION PRACTICES"  
MIL-STD-483, "CONFIGURATION MANAGEMENT PRACTICES"  
MIL-STD-490, "SPECIFICATION PRACTICES"

- THE PRODUCT SPECIFICATION IS THE BASIC DESIGN DOCUMENT FOR SOFTWARE (ALSO KNOWN AS TYPE C5 SPEC, PART 2 SPEC, OR PRODUCT BASELINE)
- THE DRAFT PRODUCT SPECIFICATION REVIEWED AT CDR IS A "CODE-TO" SPECIFICATION. IT CONTAINS DETAILED MODULE AND DATA BASE DESCRIPTIONS AND FLOW CHARTS.
- THE FINAL PRODUCT SPECIFICATION IS AN "AS-CODED" SPECIFICATION, AND IS CONSTRUCTED BY APPENDING THE COMPUTER PROGRAM LISTING TO THE DRAFT PRODUCT SPECIFICATION
- OUTLINE FOR THE PRODUCT SPECIFICATION IS WIDELY USED AND WOULD BE APPROPRIATE FOR NASA SYSTEMS



**SPECIFICATION OUTLINES (MIL-STD 490)**

PARA- GRAPH	TYPE A SYSTEM	TYPE B5 CP DEVELOPMENT	TYPE C5 CP PRODUCT
1	SCOPE	SCOPE	SCOPE
2	APPLICABLE DOCUMENTS	APPLICABLE DOCUMENTS	APPLICABLE DOCUMENTS
3	REQUIREMENTS	REQUIREMENTS	REQUIREMENTS (TECHNICAL DESCRIPTION)
3.1	SYSTEM DEFINITION	COMPUTER PROGRAM DEFINITION	FUNCTIONAL ALLOCATION DESCRIPTION
3.2	CHARACTERISTICS	DETAILED FUNCTIONAL REQUIREMENTS	FUNCTIONAL DESCRIPTION (INCLUDES FLOW CHART AND LISTING)
3.3	DESIGN AND CONSTRUCTION	ADAPTATION	STORAGE ALLOCATION
3.4	DOCUMENTATION		COMPUTER PROGRAM FUNCTIONAL FLOW DIAGRAM
3.5	LOGISTICS		
3.6	PERSONNEL AND TRAINING		
3.7	FUNCTIONAL AREA CHARACTERISTICS		
3.8	PRECEDENCE		



**SPECIFICATION OUTLINES (MIL-STD 490) (CONT.)**

<b>PARA- GRAPH</b>	<b>TYPE A SYSTEM</b>	<b>TYPE B5 CP DEVELOPEMENT</b>	<b>TYPE C5 CP PRODUCT</b>
<b>4</b>	<b>QUALITY ASSURANCE PROVISIONS</b>	<b>QUALITY ASSURANCE PROVISIONS</b>	<b>QUALITY ASSURANCE</b>
<b>4.1</b>	<b>GENERAL</b>	<b>INTRODUCTION</b>	<b>TEST PLAN/PROCEDURE CROSS-REFERENCE INDEX</b>
<b>4.2</b>	<b>QUALITY CONFORMANCE INSPECTIONS</b>	<b>TEST REQUIREMENTS</b>	<b>OTHER QUALITY ASSURANCE PROVISIONS</b>
<b>4.3</b>		<b>ACCEPTANCE TEST REQUIREMENTS</b>	
<b>5</b>	<b>PREPARATION FOR DELIVERY</b>	<b>PREPARATION FOR DELIVERY</b>	<b>PREPARATION FOR DELIVERY</b>
<b>5.1</b>			<b>PRESERVATION AND PACKAGING</b>
<b>5.2</b>			<b>MARKINGS</b>
<b>6</b>	<b>NOTES</b>	<b>NOTES</b>	<b>NOTES</b>
<b>10, 20, etc.</b>	<b>APPENDICES</b>	<b>APPENDICES</b>	<b>APPENDICES</b>



APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: PROGRAM DEVELOPMENT

REF: SAMG 4, "MONITORING AND REPORTING SOFTWARE DEVELOPMENT STATUS"  
SAMG 15, "SOFTWARE DEVELOPMENT AND MAINTENANCE FACILITIES"

- CODING STANDARDS ARE NOT ADDRESSED IN MUCH DETAIL IN THE AIR FORCE ORIENTED REGULATIONS, SPECIFICATIONS, AND STANDARDS (THE NAVY MIL-STD-1679 INCLUDES CODING STANDARDS)
- SOFTWARE DEVELOPMENT PRACTICES ARE DISCUSSED BRIEFLY IN SAMG 4:
  - TOP DOWN AND BOTTOM UP IMPLEMENTATION APPROACHES
  - MODULARITY
  - STRUCTURED OR COMPOSITE DESIGN, FUNCTIONAL DECOMPOSITION
  - REQUIREMENTS TRACEABILITY
  - STRUCTURED PROGRAMMING
  - FUNCTIONAL ORGANIZATION OF PERSONNEL

APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: TEST, EVALUATION, AND DEMONSTRATION

REF: SAMG 10, "VERIFICATION"

SAMG 11, "VALIDATION AND CERTIFICATION"

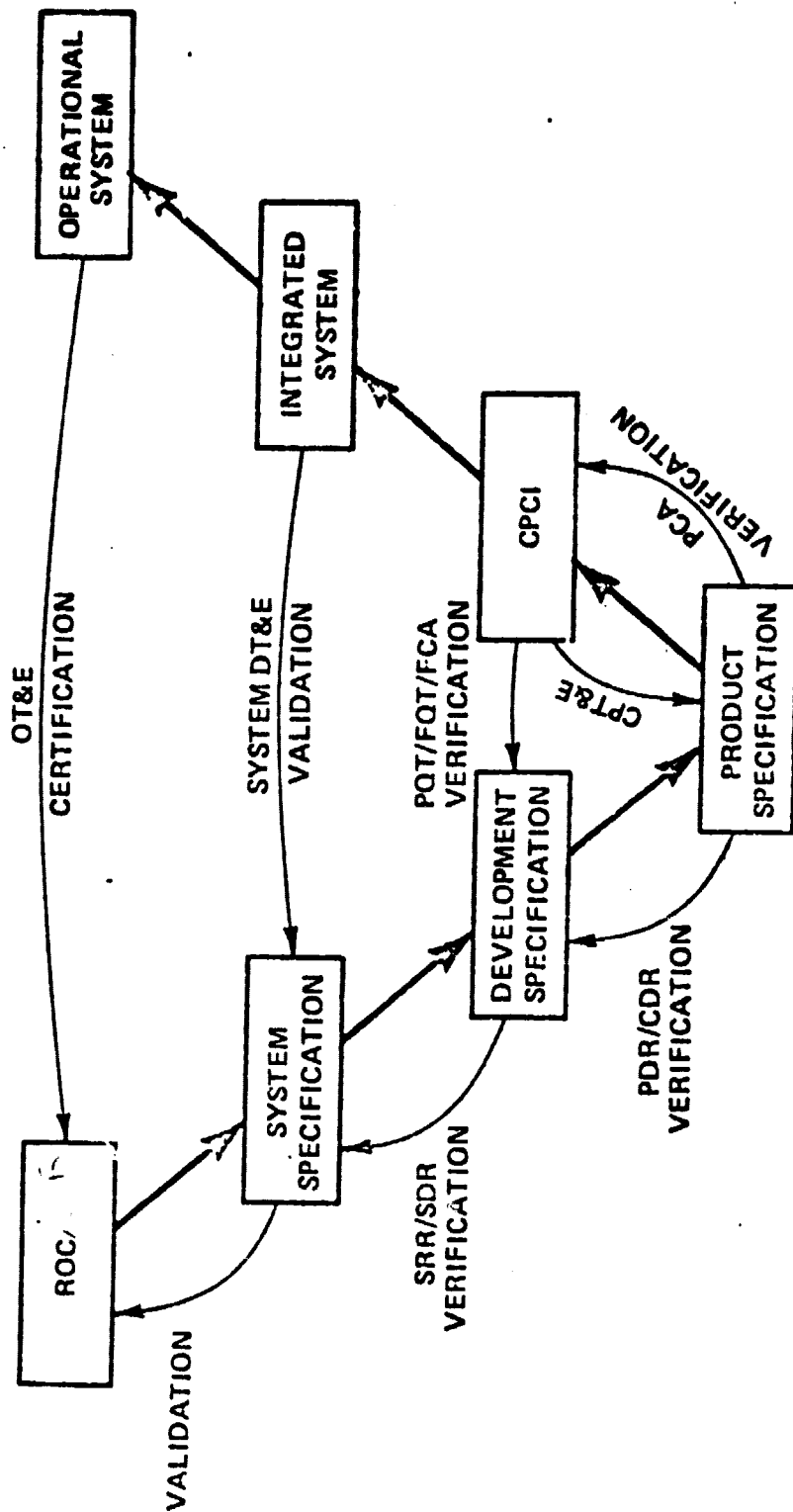
SAMG 9, "SOFTWARE DOCUMENTATION AND REQUIREMENTS"

- THE COMPUTER PROGRAM DEVELOPMENT SPECIFICATION IS THE BASIS FOR ALL PROGRAM TESTING:
  - SECTION 3 CONTAINS CPCI REQUIREMENTS
  - SECTION 4.2 CONTAINS TEST REQUIREMENTS
- BASIC TEST DOCUMENTS ARE:
  - CPCI TEST PLANS/PROCEDURES
  - CPCI TEST RESULTS
  - SYSTEM TEST PLANS/PROCEDURES
  - SYSTEM TEST RESULTS
- THESE DOCUMENTS ARE EMBEDDED IN A FORMAL TEST AND EVALUATION PROCESS THAT INVOLVES:
  - COMPUTER PROGRAM TEST AND EVALUATION (CPT&E)
  - PRELIMINARY AND FORMAL QUALIFICATION TESTS (PQT, FQT)
  - SYSTEM DEVELOPMENT TEST AND EVALUATION (DT&E)
  - OPERATIONAL TEST AND EVALUATION (OT&E)
- A SIMILAR SET OF DOCUMENTS AND MILESTONES WOULD BE USEFUL FOR NASA DEVELOPMENTS





SCOPE OF TEST AND EVALUATION



TEST DOCUMENTATION OUTLINES (DI-T-3703A)

A. CPCI TEST PLAN

1. PURPOSE
2. REFERENCES
3. TEST CONCEPTS
4. QUALIFICATION REQUIREMENTS AND CRITERIA
5. QUALIFICATION OBJECTIVES/TEST BASE SUMMARY
6. DT&E CPCI QUALIFICATION TEST IMPLEMENTATION
7. CONTROL AND REPORTING PROCEDURES

B. CPCI TEST PROCEDURE

FOR EACH QUALIFICATION TEST:

1. CAPTION
2. LOCATION AND SCHEDULE
3. REFERENCES
4. TEST OBJECTIVES
5. MANNING AND RESPONSIBILITIES
6. EQUIPMENT AND COMPUTER PROGRAM REQUIREMENTS
7. TEST OPERATING PROCEDURES
8. DETAILED TEST DESCRIPTION
9. DATA REDUCTION AND ANALYSIS



## VERIFICATION ACTIVITIES

### REQUIREMENTS VERIFICATION

- SRR, SDR
- EVALUATION OF CPDP
- AUTHENTICATION OF DEVELOPMENT SPECIFICATION
- REVIEW OF TEST PLANS

### DESIGN VERIFICATION

- PDR, CDR
- REQUIREMENTS TRACED TO INDIVIDUAL COMPUTER PROGRAM COMPONENTS
- REVIEW OF INTERFACES
- REVIEW OF TEST PROCEDURES

### COMPUTER PROGRAM VERIFICATION

- CONTRACTOR INTERNAL TESTING
- TOP-DOWN vs BOTTOM-UP INTEGRATION PHILOSOPHY
- QUALIFICATION TESTS: POT, FOT



APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: END-PRODUCT USE

REF: SAMG 9, "SOFTWARE DOCUMENTATION REQUIREMENTS"  
DID-M-3410, "USERS MANUAL (COMPUTER PROGRAM)"

- USER DOCUMENTATION (USERS MANUALS OR POSITIONAL HANDBOOKS) WOULD NORMALLY BE REVIEWED IN DRAFT FORM AT THE FUNCTIONAL CONFIGURATION AUDIT AND IN FINAL FORM AT THE PHYSICAL CONFIGURATION AUDIT
- STANDARD FORMATS ARE DESIRABLE TO FACILITATE DOCUMENT USE AND MAINTENANCE
- IT WOULD NORMALLY BE NECESSARY TO TAILOR THE STANDARD FORMAT TO THE NEEDS OF THE SPECIFIC PROGRAM



## USERS MANUAL OUTLINE (DID-M-3410)

1. INTRODUCTION
2. GLOSSARY
3. COMPUTER PROGRAM SYSTEM CAPABILITIES
  - 3.1 PURPOSE
  - 3.2 GENERAL DESCRIPTION
  - 3.3 FUNCTIONS PERFORMED
4. FUNCTION DESCRIPTION FOR EACH FUNCTION:
  - 4.1 TITLE OF FUNCTION
  - 4.2 DESCRIPTION OF FUNCTION
5. USAGE INSTRUCTIONS FOR EACH FUNCTION:
  - 5.1 PREPARATION OF INPUTS
  - 5.2 RESULTS OF OPERATION
6. OPERATING INSTRUCTIONS
  - 6.1 OPERATING PROCEDURES
  - 6.2 OPERATOR INPUTS
  - 6.3 OPERATOR OUTPUTS
7. APPENDICES

APPLICABILITY AND DOD STANDARDS AND GUIDELINES

AREA: MAINTENANCE

REF: SAMG 12, "SOFTWARE MAINTENANCE"

SAMG 15, "SOFTWARE DEVELOPMENT AND MAINTENANCE FACILITIES"

- MAINTENANCE IS NOT ESSENTIALLY DIFFERENT FROM DEVELOPMENT:  
IT REFERS TO CORRECTIONS AND MODIFICATIONS OF THE SOFTWARE PERFORMED  
AFTER END-PRODUCT ACCEPTANCE (PROGRAM MANAGEMENT RESPONSIBILITY TRANSFER).
- FACTORS SUPPORTING SOFTWARE MAINTAINABILITY

FACTOR

MAINTAINABILITY ISSUES

- |  |  |
|--|--|
| 1. PERSONNEL, FACILITIES, AND TOOLS              | PLAN FOR MAINTENANCE (CRISP)<br>ACQUIRE NEEDED FACILITIES<br>INCLUDE NECESSARY SUPPORT SOFTWARE<br>AS DELIVERABLES |
| 2. DOCUMENTATION AND CONFIGURATION<br>MANAGEMENT | ACQUIRE DOCUMENT MAINTENANCE AND<br>CONFIGURATION MANAGEMENT TOOLS<br>(INCLUDING PROGRAM SUPPORT LIBRARY)          |
| 3. INHERENT DESIGN AND UNDERSTANDABILITY         | EMPHASIS ON MAINTAINABILITY IN PLANS,<br>SPECIFICATIONS, REVIEWS, AND AUDITS                                       |
- DOD GUIDEBOOK (SAMG 12) PROVIDES A USEFUL CHECKLIST

APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: DATA COLLECTION

REF: SANG 14, "SOFTWARE COST ESTIMATING AND MEASURING"

- IT IS RECOMMENDED THAT COST DATA BE COLLECTED USING THE PROGRAM WORK BREAKDOWN STRUCTURE.
- THE SAM GUIDEBOOK SERIES DOES NOT MAKE SPECIFIC RECOMMENDATIONS REGARDING THE COLLECTION OF EITHER PRODUCTIVITY OR ERROR DATA.

APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: PROCUREMENT

REF: SAMG 3, "CONTRACTING FOR SOFTWARE ACQUISITION"  
SAMG 4, "MONITORING AND REPORTING SOFTWARE DEVELOPMENT STATUS"

SAMG 5, "STATEMENT OF WORK PREPARATION"

SAMG 16, "LIFE CYCLE EVENTS"

- ACQUISITION PROCESS FOR SOFTWARE FOLLOWS THE SAME BASIC STEPS AS THE ACQUISITION PROCESS FOR HARDWARE.
- ARMED SERVICES PROCUREMENT REGULATIONS (ASPR) ARE THE BASIS FOR ALL CONTRACTING BY DOD.
- MOST SOFTWARE DEVELOPMENT CONTRACTS ARE ISSUED ON A COST-REIMBURSEMENT BASIS (PLUS FIXED FEE, AWARD FEE, OR INCENTIVE FEE)
- THE B5 SPECIFICATION (REQUIREMENTS) IS A CRITICAL PART OF THE PROCUREMENT PACKAGE FOR SOFTWARE DEVELOPMENT; IF IT DOESN'T EXIST, CONSIDER A SEPARATE REQUIREMENTS DEVELOPMENT CONTRACT.





## **PROCUREMENT PLANNING**

### **DECISION ISSUES IN SOFTWARE PROCUREMENT**

#### **STRUCTURE OF CONTRACTOR RELATIONSHIPS**

- PRIME ONLY (HARDWARE AND SOFTWARE)**
- PRIME (HARDWARE) PLUS SUB(SOFTWARE)**
- HARDWARE CONTRACTOR, SOFTWARE CONTRACTOR, INTEGRATION CONTRACTOR**

#### **CONTRACT SOFTWARE DELIVERABLES**

- FEASIBILITY STUDY, PRELIMINARY DESIGN, DETAILED DESIGN, OPERATING SOFTWARE, MAINTENANCE SUPPORT**
- ONE CONTRACT OR SEVERAL?**

#### **PARALLEL DEVELOPMENT**

- PARALLEL SOFTWARE DESIGN OR DEVELOPMENT**
- PARALLEL HARDWARE/SOFTWARE DEVELOPMENT; SOFTWARE FIRST?**



## PROCUREMENT PLANNING (CONT.)

### THE BID PACKAGE

#### ELEMENTS OF A BID PACKAGE

- SCHEDULE: DELIVERABLE ITEMS & DATES
- STATEMENT OF WORK (SOW)
- CONTRACT DATA REQUIREMENTS LIST (CDRL): DOCUMENTATION PLUS SOFTWARE ITSELF  
(NOTE: AF DATA ITEM DESCRIPTION (U) E-129 FOR SOFTWARE)
- SPECIFICATION OR TECHNICAL EXHIBIT (SYSTEM OR SOFTWARE SPECIFICATION)
- SOLICITATION INSTRUCTIONS

#### RECOMMENDATIONS

- REQUIRE SOFTWARE (SOURCE PROGRAMS?) AND SOFTWARE DOCUMENTATION DELIVERABLES
- REQUIRE A COMPUTER PROGRAM DEVELOPMENT PLAN (CPDP)
- NEGOTIATE TO RIGHTS FOR NECESSARY SUPPORT SOFTWARE AND DOCUMENTATION



## APPLICABILITY OF DOD STANDARDS AND GUIDELINES

---

AREA: CERTIFICATION

REF: SAMG 11, "VALIDATION AND CERTIFICATION"

- THE DOD DEFINES CERTIFICATION TO MEAN "THE USING COMMAND'S AGREEMENT, AT THE CONCLUSION OF OPERATIONAL TEST AND EVALUATION (OT&E), THAT THE ACQUIRED SYSTEM SATISFIES ITS INTENDED OPERATIONAL MISSION."
- CERTIFICATION MARKS THE BEGINNING OF THE DEPLOYMENT PHASE; IT IS THE RESPONSIBILITY OF THE USING COMMAND BUT REQUIRES DEVELOPMENT COMMAND SUPPORT.
- THE DOD GUIDEBOOKS DO NOT ADDRESS CERTIFICATION IN THE USUAL SENSE OF THE WORD.\*

---

\*"CERTIFICATION IS AN AUTHORITATIVE ENDORSEMENT OF THE CORRECTNESS OF A PROGRAM,"  
G.J. MYERS, SOFTWARE RELIABILITY



Survey IntervieweesJSC/Shuttle

R. Parten - Chief, Spacecraft Software Div.  
J. Stokes - Chief, Ground Data Systems Div.  
A. Aldrich - Manager, Orbiter Avionics Systems Office  
L. Dunseith - Director, Data Systems and Analysis Div.  
J. Aaron - Deputy Chief, Spacecraft Software Div.

MSFC/Space Telescope

J. T. Powell - Director, Data Systems Lab.  
W. C. Bradford - Deputy Director, Data Systems Lab  
D. Aichele - Manager, Software Engineering Div.  
C. Ballantine - Space Telescope, Software Manager

GSFC/Landsat D

L. Green - Landsat D, NOS Projects  
W. Webb - Landsat D, Data Engineer  
F. McGarry - Software Engineering Lab  
D. Krueger - Chief, Systems Div.  
T. Taylor - Systems Div.

JPL/Galileo

A. Irvine - Manager, Data Processing and Management Science  
R. Loesh - Galileo, Software Manager  
B. Larman - Galileo, Spacecraft Software System Engineer  
P. Molko - Galileo, Ground Software System Engineer

TRW

B. Boehm - Director, Software Research and Technology

System Development Corp.

J. Munson - V.P., Corporate Software Engineering  
T. Court - Software Development Manager

Air Force - ESD

J. Grewe - Lt Col., Manager, Software Acquisition Guidebook Series

A-2     Software Management Workshop - Prospective Participants\*

- .     JSC - Jim Stokes (Landsat D Audit Chairman)  
             Dick Parten (Chief-Spacecraft Software Div)
- .     MSFC - James Powell (Director, Data Systems Lab)  
             Cliff Bradford (Deputy Director)
- .     GSFC - Lloyd Green (Landsat D, NOS Projects)  
             Frank McGarry (Software Engineering Lab)
- .     JPL - Bob Loesh (Galileo Software Manager)  
             Gentry Lee (Co-Author of NMI 2410)
- .     KSC - Walt Murphy (Shuttle Program)
- .     LaRC - Ed Fournier (MUST/NEEDS Programs)
- .     TRW - Barry Boehm (Director - Software Research & Technology)
- .     SDC - Jack Munson (V. P. Corporate Software Engineering)
- .     Air Force - Lt. Col. John Grewe (responsible for ESD software  
             acquisition guidebook series)
- .     NASA Hdqtrs - Bill McInnis - (Study Manager)  
                     Harry Sonnemann (Deputy Chief Engineer)  
                     Leonard Jaffe (Special Assistant to Chief Engineer)
- .     Bell Labs - Gordon Heffron (Director, A.F. Woods Hole Study)
- .     Consultant - Bill Tindall (Space Telescope Audit Chairman)
- .     CSDL - Malcolm Johnston (Study Manager)
- .     CSDL - Bart DeWolf             (Study Team)
- .     CSDL - Jim Kernan             (Study Team)
- .     CSDL - Bob O'Donnell         (Study Team)
- .     CSDL - Lance Drane           (Study Team)

\* One representative is expected from each Center and in some cases two, though our desire is to keep the total participants to approximately 20.

- Interview Format -Purpose of Study

The objectives of this study are to refine NASA-wide understanding of flight software development processes and problems and, as a result, develop management policy improvement recommendations that can be widely agreed to within the : Implementation Centers.

Study Approach

These survey interviews augment our previous review of pertinent policy documentation to round out our understanding of the present software practitioners' views.

Seven study-subjects have been selected for this review: JSC (Shuttle), MSFC (ST), GSFC (Landsat D), JPL (Galileo), TRW, SDC, and the Air Force - ESD.

The purpose of the review is to enable a compilation and generic evaluation of current software management practices. Individual study-subject data will not be cited in the resulting recommendations or reports without the expressed approval of the study-subject organization.

The present plan is to convene a NASA peer group to review a preliminary draft of recommendations, prior to submission, to insure acceptability amongst these key Center software development personnel.

Scope of Interview

This interview format is intended to provide some focus and direction to these informal exchanges. Two categories of questions have been generated: those that are generic to the software development process (enclosed) and those that are specifically related to the study-subject's policies, procedures, etc. Many of these questions may be inappropriate for any single interviewee.

### Purpose of Interview

- o Determine what policies, practices, procedures, etc. you find useful and not so useful.
- o Solicit your thoughts for improving same, including potentially counter-productive changes
- o Solicit your thoughts on the adequacy of present Headquarters and Center policies (eg. NMI 2410.6)
- o Solicit suggestions on improving our study approach, including the planned Workshop.

### Generic Questions

- 1.) From your perspective, what has been the most difficult, troublesome aspect of software development?
  - o Stable initial system design?
  - o Software requirements generation? specification? verification?
  - o Evolving/changing requirements?
  - o Software design? coding? verification?
  - o Software testing? S/H integration & compatibility testing?  
Overall system testing?
  - o Tool development? Test & other facility development?
  - o Configuration management?
  - o Organizational matters? contractual matters?
  - o Cost &/or schedule estimation & control?
  - o Risk assessment? managerial visibility?
  - o Other?
  
- 2.) How do these development difficulties compare with difficulties encountered during operational &/or maintenance phases?
  
  
  
  
  
  
  
  
  
  
- 3.) In retrospect, what could have been done to ease the problems cited above?
  - o Better plan?
  - o More (or less) standards or guidelines?
  - o More/better resources?
  - o Greater discipline (e.g., via top-level policy or management support)?
  - o other?



4.) Which aspects of the software development cycle have been the least troublesome? Why?

5.) How effective were the policy documents your project used?

a.) Did they accurately reflect your actual policies, practices, procedures?

b.) If not, does this suggest the need for better policies? fewer policies? greater enforcement of present policies?

c.) Where is your policy documentation inaccurate? incomplete? in conflict with contractor policies or preferences?

d.) Is your present policy stated at the correct level (e.g., too little or too much specificity)? extent (e.g., RFP thru contract award, development, operations, maintenance, improvements, etc)?

- e.) What present policies, or potential "improvements", would in your view be counter-productive?
  - f.) Would you apply this same policy documentation to your next project? big or small? internal or contracted? complex or straightforward?
  - g.) Assuming deficiencies you have identified were corrected, would you recommend this documentation as a NASA-wide policy source?
  - h.) Who generated these policies? are they being updated or refined? How does this process work? Is the documentation we reviewed complete? up-to-date?
- 6.) Do (did) you find NMI 2410.6 helpful? How could it be improved? Is the audit process useful? Would more of these institutionalized policies or guidelines be helpful? Fewer?

- 7.) What does the implementation of the idealized form of the various policies, standards, or steps cost? How are these costs estimated? Which proportionally seem to be most cost-effective?
- 8.) What software procurement standards or procedures does the organization now have in place? At what level are they implemented? (eg. NASA Hdqtrs. vs Center-level)
- 9.) Would separately contracted life cycle phases (eg. requirements specifications and design specifications) improve software quality? What problems might this create?
- 10.) What mechanisms are provided to ensure that contractor software development utilizes your policies (or equivalent)?
- 11.) How do in-house vs. contracted management policies and organizations differ? What is the mix in your organization?

12.) How do the software elements of a procurement RFP get generated?

13.) How do Headquarters-level vs. project-level management policies and organizations differ?

14.) How will future digital processing system architectural or technological advances affect present management policies, practices, etc.?

15.) How could the study approach, the interviews, or the workshop be improved?

16.) What questions should I have asked that I didn't?

#### Specific Questions

Listed subsequent to completion of study-subject's documentation review.

Areas of Concern During  
Software Development  
- A Checklist -

<u>Area</u>	<u>Possible Needs</u>
<b>A. <u>Essential for Management</u></b>	
1.) Top-Level Policy	o Statement of objectives and supporting policy, including required procedures (items below)
2.) Management Plan	o Guidelines for required content (incl. scope; organization; roles and responsibilities; relationship between organizations; authority assignments; life cycle phase summary; status reporting; schedule, cost, & change control; available documentation).
3.) Configuration Management	o Guidelines for implementing & documenting plans (incl. items controlled; degree of control; phasing; authority assignments; status accounting; verification); I-Load roles, responsibilities, & techniques.
4.) Quality Assurance	o Quality measures; guidelines for implementing & documenting plans (incl. organizational responsibilities & authority; reviews & audits)
5.) End-Product Acceptance	o Performance & quality measures; guidelines for implementing & documenting plans, reviews, waivers, & acceptance.
6.) Operations & Maintenance Plan	o Guidelines for generating, reviewing, documenting & updating plan over total life-cycle.
7.) Resource & Schedule Estimation, Monitoring, & Control	o Suggested techniques; accuracy assessment; authority assignments; guidelines for implementing and documenting plans.
8.) Risk Assessment	o Suggested techniques; required reporting.
9.) Documentation	o Guidelines for documents required (incl. content; format; scheduling; updating; distribution).

-Area

Possible Needs

B.) Essential for Development & Use

10.) Development Plan

- o Guidelines for generating, reviewing, documenting & updating plan (incl., engineering methodology; critical path monitoring techniques, eg., PERT; standard life-cycle event milestones & nomenclature, eg., DRR, PDR, CDR; procedures for reviews & disposition of discrepancies; S/H integration approach).

11.) Requirements Specifications

- o Guidelines for generating, reviewing, documenting & updating specs, (incl. opert'l., system, interface, flight software, and development & test facility levels; performance, quality, cost & schedule goals; test reqmts; acceptance criteria).

12.) Design Specifications

- o Guidelines for generating, reviewing, documenting & updating specs (incl. preliminary & detailed levels; design methodology options; traceability to corresponding reqmts).

13.) Program Development

- o Design, coding, & tool standards (incl. Programmers Handbook); techniques & methodologies; support software & facilities; guidelines for unit & system-level development, integration, & control; resource management and progress reporting.

14.) Verification, Validation  
Testing, Evaluation &  
Demonstration

- o Techniques, tools, support software & facilities; guidelines for generating, reviewing, documenting, & updating plans (incl. opert'l, system, interface, & software levels; results review/analysis; disposition of discrepancies; relationship of test to design methodologies; verification against design & reqmts; S/H compatibility testing; post-flight software evaluation; IV&V)

- 15.) End-Product Use
    - o Guidelines for generating, reviewing & updating Users, Support, & Training Manuals.
  - 16.) Maintenance
    - o Guidelines for participation in reqmts. specification; operational criteria & procedures.
- C.) Specialized Needs
- 17.) Data Collection
    - o Standard measurements & reporting forms (eg., for productivity & error data); data collection cost impact estimation.
  - 18.) Procurement
    - o Standard nomenclature, procedures, & outlines (eg., for SOW, WBS, etc.); guidelines for generating RFP's & reviewing responses; definition of deliverables (incl. documentation, flight & support software).
  - 19.) Certification
    - o Performance & quality measures; guidelines for implementing and documenting plans; outline of legal aspects.